

Oracle® Enterprise Manager Cloud Control

IBM Tivoli Netcool/OMNIBus Connector Installation and Configuration Guide



For Enterprise Manager 13c Release 5

F37150-02

July 2022

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

F37150-02

Copyright © 2015, 2022, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Audience	vii
Documentation Accessibility	vii
Related Documents	vii
Conventions	viii

1 Introduction to the Connector

Versions Supported	1-1
Retry	1-1

2 Installing the Connector

Installation Platforms	2-2
Prerequisites	2-2
Installing the Connector in Enterprise Manager	2-3
Installing the Connector if Enterprise Manager is in "Offline" Mode	2-4
Exporting the Installation Files	2-5
Installing the Web Service for Netcool/OMNIBus Front-End	2-6
Web Service for Netcool/OMNIBus on Unix	2-6
Prerequisites	2-7
Installing the Web Service on Unix	2-7
Running the Web Service on Unix	2-9
Testing the Web Service on Unix	2-9
Copying the Required .jar Files	2-10
Uninstalling the Web Service on Unix	2-11
Web Service for Netcool/OMNIBus on Windows	2-11
Prerequisites	2-12
Installing the Web Service on Windows	2-12
Running the Web Service as a Windows Service (Optional)	2-14
Running the Web Service on Windows	2-14
Testing the Web Service on Windows	2-15
Copying the Required .jar Files	2-15

Uninstalling the Web Service on Windows	2-16
Installing the Oracle Agent for Netcool/OMNIbus	2-17
Uninstalling the Oracle Agent for Netcool/OMNIbus	2-18
Modifying Netcool/OMNIbus	2-18
Configuring the Object Server Database	2-19
Configuring the TEC EIF Probe	2-19
Change Deduplication Trigger Priority	2-20
Uninstalling Modifications to Netcool/OMNIbus	2-21
Uninstalling the Connector	2-22

3 Configuring the IBM Tivoli Netcool/OMNIbus Connector

Creating a Connector Instance	3-1
Configuring the Connector	3-2
Enabling and Disabling the Connector	3-3
Providing General Settings	3-4
Connection Settings	3-4
Retry	3-5
Testing the IBM Tivoli Netcool/OMNIbus Connector	3-5

4 Forwarding Events to IBM Netcool/OMNIbus

Setting Event Rules	4-1
Viewing Events in the IBM Netcool/OMNIbus Console	4-3

5 Working with Event Templates

Default Templates	5-1
Create Event Template	5-2
Update Event Template	5-2
Details Mapping - Create	5-3
Details Mapping - Update	5-4
Journal Mapping	5-4
Severity Mappings	5-4
Customizing Event Mappings	5-5
XML Format of Oracle Enterprise Manager Events	5-5
Managing Templates in Oracle Enterprise Manager 12c	5-6
Edit a Template with Enterprise Manager	5-7
View Template Code in Enterprise Manager	5-7
Delete a Template with Enterprise Manager	5-7
Restore a Deleted Template with Enterprise Manager	5-8
Manually Changing a Mapping	5-8

Exporting the Template Files	5-8
Modifying Templates	5-9
Registering Templates	5-10
Template Customization Example	5-10

6 Enabling SSL for HTTPS

Configure Oracle Enterprise Manager to Use SSL	6-1
Installing a Self-Signed Certificate	6-1
Installing a Certificate from a Certified Authority	6-2
Importing the Web Service Certificate into Enterprise Manager	6-3

7 Troubleshooting

Diagnosing the Problem	7-1
Troubleshooting Web Service Startup Errors	7-3
JVM Errors	7-4
Using the Correct URL for Web Service Operations	7-5
Determining the URL Used by the Web Service	7-5
Troubleshooting Web Service Operations Errors	7-5
Resolving Errors from Oracle Enterprise Manager	7-6
Invalid Web Service Credentials	7-7
SSL Not Configured in Enterprise Manager	7-8
Web Service is Down	7-8
Invalid IP Address	7-9
Invalid URL Path	7-9
Firewall Blocking Access	7-10
Invalid Port Number	7-10
Unknown Host	7-11
Invalid XML Format	7-11
Missing Log Jar	7-12
Timeout	7-12
Cannot Connect to the Probe	7-14

A Web Service for Netcool/OMNIbus Details

XML Format of Netcool/OMNIbus Alerts	A-1
Mappings Between XML Format and Message Field Names	A-2
Log Entries	A-3
Extended Fields	A-3
Changing Default Port Numbers	A-4
Changing the Web Service Port (8080)	A-5

Changing the Web Service Port (8443)	A-6
Changing Web Service Credentials	A-6

B Enterprise Manager Event Format

EMEvent.xsd	B-1
connectorCommon.xsd	B-3
externalEvent.xsd	B-9

C Web Service for Netcool/OMNIBus EventService WSDL Format

D Enterprise Manager Event Examples

Create Example	D-1
Update Example	D-4
Clear Example	D-7

Preface

This *IBM Tivoli Netcool/OMNIBus Connector Installation and Configuration Guide* provides the information that you require to install and configure the IBM Tivoli Netcool/OMNIBus Connector that integrates Oracle Enterprise Manager with IBM Tivoli Netcool/OMNIBus management tools.

Audience

This guide is written for Oracle Enterprise Manager system administrators who want to install and configure the IBM Tivoli Netcool/OMNIBus Connector to enable integration between Oracle Enterprise Manager and IBM Tivoli Netcool/OMNIBus.

You should already be familiar with Oracle Enterprise Manager.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Documents

For more information, see the following books in the Oracle Enterprise Manager documentation set:

- *Oracle Enterprise Manager Cloud Control Basic Installation Guide*
- *Oracle Enterprise Manager Cloud Control Advanced Installation and Configuration Guide*
- *Oracle Enterprise Manager Cloud Control Administrator's Guide*
- *Oracle Enterprise Manager Cloud Control Upgrade Guide*
- *Oracle Enterprise Manager Framework, Host, and Services Metric Reference Manual*
- *Oracle Enterprise Manager Command Line Interface*
- *Oracle Enterprise Manager Cloud Control Extensibility Programmer's Guide*
- *Oracle Database 2 Day DBA*

The latest versions of this and other Oracle Enterprise Manager documentation can be found at: <https://docs.oracle.com/en/enterprise-manager/cloud-control/enterprise-manager-cloud-control/13.4/index.html>

Oracle Enterprise Manager also provides extensive online help. Click **Help** on any Oracle Enterprise Manager page to display the online Help system.

To download free release notes, installation documentation, white papers, or other collateral, please visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at: <http://otn.oracle.com/membership/>

If you already have a user name and password for OTN, then you can go directly to the documentation section of the OTN Web site at: <http://otn.oracle.com/documentation/>

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Introduction to the Connector

The IBM Tivoli Netcool/OMNIBus Connector integrates Oracle Enterprise Manager with IBM Tivoli Netcool/OMNIBus through web services, enabling the sharing of event information from Oracle Enterprise Manager to IBM Tivoli Netcool/OMNIBus.

The connector monitors all events sent from Oracle Enterprise Manager and automatically updates alert information in Netcool/OMNIBus. The connector supports the following features:

- Sharing of event information from Oracle Enterprise Manager to IBM Tivoli Netcool/OMNIBus.
- Customization of event to alert mappings between Oracle Enterprise Manager and IBM Tivoli Netcool/OMNIBus.
- Synchronization of event changes in Oracle Enterprise Manager with the alerts in IBM Tivoli Netcool/OMNIBus.

Versions Supported

This connector supports the following versions of Oracle Enterprise Manager and IBM Tivoli Netcool/OMNIBus:

- Oracle Enterprise Manager Cloud Control 12c Release 1 (12.1.0.1) and higher versions.
- IBM Tivoli Netcool/OMNIBus 7.2, 7.2.1, 7.3, 7.3.1, 7.4, and 8.1.

The base Enterprise Manager version number for IBM Tivoli Netcool/OMNIBus Connector Release 12.1.0.2 is Enterprise Manager 12c Release 1 (12.1.0.1) and higher versions.

Retry

The “Retry” section, available on the connector configuration page, enables you to specify whether a failed ticketing request needs to be retried within a configurable expiration time.

Enabling the Retry option gives you the option of specifying whether you want to resend a ticketing request if the request fails the first time, and specifying the time period after which you want to abandon the retry. Enterprise Manager retries every 2 minutes until the request is successful or the retry interval expires.

2

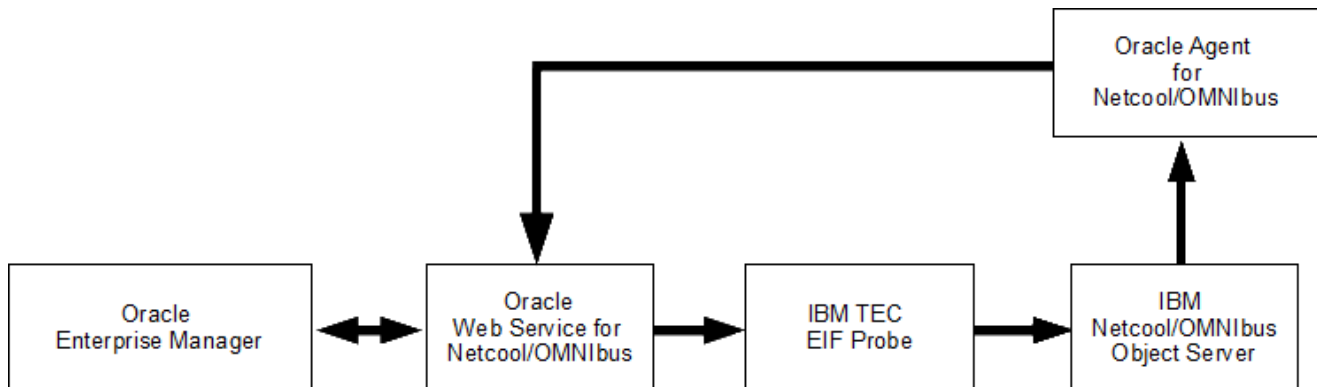
Installing the Connector

The Oracle Enterprise Manager Connector Framework requires a web service interface for forwarding event information to IBM Tivoli Netcool/OMNIBus (Netcool/OMNIBus). Since Netcool/OMNIBus does not provide a web services interface, you must install a third-party web service for Netcool/OMNIBus front-end, which is included in the Oracle Enterprise Manager Netcool/OMNIBus installation package.

You can install the web service on a Windows or Unix system that has connectivity to the Netcool/OMNIBus server. In addition to the web service for Netcool/OMNIBus front-end, you must also install a back-end Oracle Agent for Netcool/OMNIBus on the same physical system as the Netcool/OMNIBus server. The Oracle Agent for Netcool/OMNIBus is preconfigured and is also included in the Oracle Enterprise Manager IBM Tivoli Netcool/OMNIBus installation package.

Figure 2-1 shows the communication between the various components of the Netcool/OMNIBus Connector.

Figure 2-1 Connector Communication Between Components



The following sections in this chapter discuss these topics:

- [Installation Platforms](#)
- [Prerequisites](#)
- [Installing the Connector in Enterprise Manager](#)
- [Installing the Web Service for Netcool/OMNIBus Front-End](#)
- [Web Service for Netcool/OMNIBus on Unix](#)
- [Web Service for Netcool/OMNIBus on Windows](#)
- [Installing the Oracle Agent for Netcool/OMNIBus](#)
- [Modifying Netcool/OMNIBus](#)
- [Uninstalling the Connector](#)

Installation Platforms

You can install the web service for IBM Tivoli Netcool/OMNIBus on the following platforms that support Java Development Kit (JDK):

- Microsoft Windows
- Oracle Solaris
- Linux

You can install the Oracle Agent for IBM Tivoli Netcool/OMNIBus on the following platforms:

- Microsoft Windows (2000, 2003, 2008, XP)
- IBM AIX (RS/6000 OS version 5.2+, 5.3, 6.x, and 7.x)
- Linux RHEL (64-bit) 5.3, 5.4, 5.6
- Oracle Solaris (8, 9, and 10)
- SUSE 12 (64-bit)*

*Enterprise Manager version 13.2.0.0.0 and above supports SUSE 12 with IBM Tivoli Netcool/OMNIBus Connector Release 12.1.0.3.0 and above.

Prerequisites

Ensure that the following prerequisites have been met before proceeding to the next section.

- My Oracle Support Preferred Credentials must be set to enable you to download the Connector from the Self Update page of the Enterprise Manager Cloud Control.

To set the credentials:

1. From the Setup menu of Enterprise Manager Cloud Control, select **My Oracle Support**, then select **Set Credentials**.
2. Enter the user name and password information, then click **Apply**.
3. From Setup menu, select **Extensibility**, then **Self Update**.
4. From the Self Update page, click **Check Updates** to schedule a job to collect the metadata for the connector.

See Also:

"Using Self Update to Download Oracle Products" section of the *Enterprise Manager Cloud Administration Guide* for more details.

- An IBM Tivoli Netcool/OMNIBus user account must be set up that the web service for Netcool/OMNIBus can use to access the Object Server database. The account must have read permission to the `alerts.status`, `alerts.details`, and `alerts.journal` database tables.

- An unzip utility must be available on the system where the Oracle Agent for Netcool/OMNIbus is to be installed.
- A Netcool/OMNIbus Process Agent must be installed on the system where the Oracle Agent for Netcool/OMNIbus is to be installed. Oracle recommends installing the Oracle Agent for Netcool/OMNIbus on the Object Server system.
- The Netcool/OMNIbus connector requires a Netcool/OMNIbus TEC EIF Probe to insert events into IBM Tivoli Netcool/OMNIbus. This probe can be a single standalone master or a master/slave pair. At a minimum, a master Netcool/OMNIbus TEC EIF Probe must be installed on a system that has connectivity to the Object Server system.

Installing the Connector in Enterprise Manager

The Netcool/OMNIbus connector is not included out-of-box, so you need to download it from the Oracle Enterprise Manager store. All connectors in the store are available for download via the Self Update page of Enterprise Manager Cloud Control.

Note:

- There are no special privileges needed to download a connector. However, you need the Super Administrator privilege to apply a connector.
- If Enterprise Manager is not connected to My Oracle Support, then refer to [Installing the Connector if Enterprise Manager is in "Offline" Mode](#).

To download and install (apply) the connector:

1. From the Setup menu, select **Extensibility**, then select **Self Update**.
2. Click the **Management Connector** link in the Type column.
The IBM Tivoli Netcool/OMNIbus connector appears in the list with a status of Available.
3. Click on **IBM Tivoli Netcool/OMNIbus** connector to select it, then click **Download**.
The Schedule Download window appears, where you can determine when the download should be performed.
4. Click **Select** to download immediately.
If you want to schedule the download for a later time, specify the date and time when the download should occur, and click **Select**. You will need to return and finish the remainder of this procedure after the scheduled download date and time.
5. If necessary, refresh the screen until the status of the connector changes to Downloaded.
6. Click on the IBM Tivoli Netcool/OMNIbus connector to select it, and then click **Apply**.
After you respond to the prompt to confirm the operation, a page appears that indicates the request has been submitted.
7. Refresh the screen until the status of the connector changes from Apply Scheduled to Applied.

 **See Also:**

"Using Self Update to Download Oracle Products" section of the *Enterprise Manager Cloud Administration Guide* for detailed information and diagnostic procedures.

Installing the Connector if Enterprise Manager is in "Offline" Mode

Under certain circumstances, such as in high security environments, an active Internet connection between Enterprise Manager and the Enterprise Manager Update Store may not be available. In such situations, Enterprise Manager can be set to install the connector in an "offline" mode.

The installation process still requires that a computer exist at your site that has Internet access, as a connection to the Enterprise Manager Update Store is still required to obtain the necessary files. The files that you download to this computer can then be transferred to a computer behind your firewall.

To install the connector if Enterprise Manager is in "offline" mode:

1. From the system that you will ultimately deploy the connector, set Enterprise Manager Cloud Control to Offline Mode:
 - a. From the Setup menu, select **Provisioning and Patching**, then select **Offline Patching**.
 - b. In the Online and Offline Settings page, select **Offline**.
2. From the Setup menu, select **Extensibility**, then select **Self Update**.
3. On the Self Update page, click **Check Updates**. A message appears with the following URL to an Oracle site from where the updates catalog file can be downloaded:

```
https://updates.oracle.com/Orion/Download/download_patch/  
p9348486_112000_Generic.zip
```

4. From an Internet-enabled computer, download the catalog file using the aforementioned URL.
5. Copy the downloaded catalog file to the OMS host or the Management Agent host where you plan to import the connector.
6. Import the catalog file to Enterprise Manager:

- If the catalog file is on the OMS host:

```
emcli import_update_catalog -file="file" -omslocal
```

Where:

- `-file`: is the direct path to the connector archive (*.zip).
 - `-omslocal`: indicates that the path mentioned in the `-file` option is directly accessible to the Enterprise Manager server
- If the catalog file is on a different host:

```
emcli import_update_catalog -file="file" -host="hostname" [-
credential_set_name="setname"] | -credential_name="name" -
credential_owner="owner"
```

Example 2-1 shows a sample for importing the catalog archive.

7. On the Self Update page, in the table, click **Management Connectors**.
8. On the Connector Updates page, select the imported update that is available for download. Click **Download**.
A message appears with a URL to an Oracle site from where the update can be downloaded.
9. From a computer that is connected to the internet, download the update using the aforementioned URL.
10. Copy the downloaded file to the OMS host or the Management Agent host where you plan to deploy the connector.
11. To import the downloaded connector archive into Enterprise Manager, run the following command:

```
emcli import_update -file="<path to *.zip file>" -omslocal
```

Where:

- `-file`: is the direct path to the connector archive (*.zip).
- `-omslocal`: indicates that the path mentioned in the `-file` option is directly accessible to the Enterprise Manager server

Example 2-1 Sample for Importing Catalog Archive

```
emcli import_update_catalog -file="/u01/common/p9348486_112000_Generic.zip" -omslocal
```

Imports the master catalog file `p9348486_112000_Generic.zip`. The file must exist on the OMS host. In a multiple OMS setup, the request can be processed by any OMS, so the file should be accessible from the OMS processing the request. This means that the file must be kept on a shared location that is accessible from all the OMS instances.

```
emcli import_update_catalog -file="/u01/common/p9348486_112000_Generic.zip" -
host="host1.example.com" -credential_set_name="HostCredsNormal"
```

Imports the master catalog file `p9348486_112000_Generic.zip` that is present on the host `host1.example.com`. The host must be a managed host target in Enterprise Manager, and the Management Agent on this host must be up and running. The preferred unprivileged credentials for host `host1.example.com` are used to retrieve the remote file.

Exporting the Installation Files

The agent and web service installation files are included in the Self Update archive that was installed in [Installing the Connector in Enterprise Manager](#).

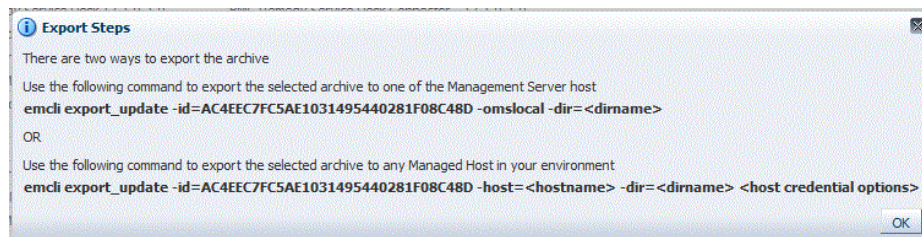
To export the installation files:

1. Determine the command required to export the adapter installation file. To do this, perform the following steps:
 - a. From the Setup menu, select **Extensibility**, then **Self Update**.
 - b. Click on the **Management Connector** link in the Type column.

- c. Click on the **IBM Tivoli Netcool/OMNIBus** connector to select it, then select **Export** from the Actions list.

A pop-up window similar to the example shown in [Figure 2-2](#) appears with the command required to export the file.

Figure 2-2 Export Command Example



2. Open a command window on the Enterprise Manager host system and navigate to a temporary directory where you can copy the file.
3. Log into EM CLI using the following command:

```
emcli login -username=<Enterprise Manager Super Administrator log in>
```

You will be asked to provide the password information.

You must execute EM CLI from the OMS server system. See the *Oracle Enterprise Manager Command Line Interface* for information about setting up EM CLI.

4. Run the EM CLI `export_update` command from your system, changing `<dirname>` to the full path of the temporary directory.

This action creates a zip file. The file name is comprised of the ID value you specify in the `export_update` command with a `.zip` extension. In the example command in [Figure 2-2](#), the zip file name would be:

```
1208A7C0D8913455B18D12F65ABCA8A0.zip
```

5. Extract the installation files from the zip file using the following command:

```
unzip *.zip archives/*
```

Installing the Web Service for Netcool/OMNIBus Front-End

The web service for Netcool/OMNIBus acts as a front-end for all data flowing into and out of Netcool/OMNIBus. Oracle Enterprise Manager posts calls to the web service whenever it needs to create or update an alert or get new or updated alerts from Netcool/OMNIBus.

You can install the web service for Netcool/OMNIBus on any Unix or Windows system that runs the Oracle JDK and has connectivity to the Netcool/OMNIBus server and the Oracle Enterprise Manager OMS (Oracle Management Service).

Web Service for Netcool/OMNIBus on Unix

The following sections explain how to install and then subsequently run the Web Service on a Unix platform:

- [Prerequisites](#)
- [Installing the Web Service on Unix](#)
- [Running the Web Service on Unix](#)
- [Testing the Web Service on Unix](#)
- [Copying the Required .jar Files](#)
- [Uninstalling the Web Service on Unix](#)

Prerequisites

The following prerequisites must be met before proceeding to the next section:

- IBM Tivoli Netcool/OMNIBus Connector Release 13.2.1.0.0 and 13.2.2.0.0 requires JDK 8 Update 5 and later or JDK 11 (any version) installed on the UNIX platform.
- IBM Tivoli Netcool/OMNIBus Connector Release 12.1.0.x.0 supports only JDK 6 and requires JDK 6 Update 11 and later.
- `JAVA_HOME` environment variable is set to the JDK installation directory.



Note:

IBM Tivoli Netcool/OMNIBus Connector does not support JDK 7, JDK 9, and JDK 10.

Installing the Web Service on Unix

To install the web service on a Unix platform, perform the following steps:

1. Create a directory where you want to install the web service.
2. Copy the `OMNIBus_webservices_adapter.jar` file from the Management Server host system to the web service installation directory. The file is located in the archives directory that was extracted in [Exporting the Installation Files](#).
3. Open a command prompt window and change the working directory to the web service installation directory.
4. Enter the following command to extract the web services components from the web services .jar file:

```
jar xvf OMNIBus_webservices_adapter.jar
```



Note:

If the system where the web service for Netcool/OMNIBus is being installed does not have the JDK installed, you cannot extract the jar file contents. You need to copy the .jar file to a system that has the installed JDK and transfer the files after they have been extracted.

This creates the `adapters` directory that contains the installation files.

5. Enter the following command to change the working directory:

```
cd adapters/endpoints/omnibus
```

6. Enter the following command to run the installation script:

```
sh ./install.sh
```

7. When the script prompts whether you want to use HTTPS:

- If you specify **Y**, then the web service is set up to use HTTPS port number 8443.
- If you specify **N**, then the web service is set up to use HTTP port number 8080.

8. When the script prompts for the user name of the web service, enter a user name that must be provided to access the web service for Netcool/OMNIBus.

The user name can be any value and is not associated with any specific OS or Netcool/OMNIBus account. Make a note of this value and supply it when configuring the Netcool/OMNIBus connector in Enterprise Manager.

9. When the script prompts for the password of the web service, enter the password that must be provided to access the web service for Netcool/OMNIBus.

The password can be any value and is not associated with any specific OS or Netcool/OMNIBus account. Note this value and supply it when configuring the Netcool/OMNIBus connector in Enterprise Manager.

10. When the script prompts for the username to access the Object Server database, enter the username of an account that has read access to the `alerts.status`, `alerts.details`, and `alerts.journals` database tables.

11. When the script prompts for the password to access the Object Server database, enter the password for the account specified in the previous step.

12. When the script prompts for the host name of the Object Server system, enter the host name or IP address of the system where the Object Server is installed. You will not be allowed to specify a host name of localhost, because the web service will fail when attempting to connect to the Object Server database.

13. When the script prompts for the Object Server database port number, enter the port number to use when connecting to the Object Server database. The default port number is **4100**. To determine the port number used by your system, start the Netcool/OMNIBus Administrator and open the Object Servers window. The port number for the Object Server is listed on this window.

14. The web service can be configured to attempt to connect to a backup Object Server database if there are problems connecting to the primary Object Server database. You are prompted to specify whether you want to configure the web service to connect to a backup Object Server instance. If you answer N to the prompt, skip to step 17.

15. When the script prompts for the host name of the backup Object Server system, enter the host name or IP address of the system where the backup Object Server is installed. You are not allowed to specify a host name of localhost, because the web service will fail when attempting to connect to the Object Server database.

16. When the script prompts for the backup Object Server database port number, enter the port number to use when connecting to the backup Object Server database. The default port number is **4100**. To determine the port number used by

your system, start the Netcool/OMNIbus Administrator and open the Object Servers window. The port number for the Object Server is listed on this window.

17. When the script prompts for the master EIF probe host name, enter the host name or IP address of the system where the master TEC EIF probe is installed.
18. When the script prompts for the master EIF probe port number, enter the port number to use when connecting to the master TEC EIF probe. The port number is defined in the `PortNumber` property in the `tivoli_eif.props` file. The file is located in the EIF probe installation directory.
19. The web service can be configured to also send events to a slave EIF probe when sending events to the master EIF probe. You will be prompted to specify whether you want to configure the web service to send events to a slave EIF probe. If you answer **N** to the prompt, skip to step 22.
20. When the script prompts for the slave EIF probe host name, enter the host name or IP address of the system where the slave TEC EIF probe is installed.
21. When the script prompts for the slave EIF probe port number, enter the port number to use when connecting to the slave TEC EIF probe. The port number is defined in the `PortNumber` property in the `tivoli_eif.props` file. The file is located in the EIF probe installation directory.
22. After the script displays the message "Netcool/OMNIbus Adapter Install Complete," press **Enter** to complete the installation.
23. Copy additional required jar files as specified in [Copying the Required .jar Files](#)
24. If the web service was configured to run using the HTTPS protocol, you must set up SSL as specified in [Configure Oracle Enterprise Manager to Use SSL](#).
25. Delete the `OMNIbus_webservices_adapter.jar` file from the installation directory.

The web service framework is now installed and ready to start.

Running the Web Service on Unix

To run the web service for Netcool/OMNIbus framework commands listed with the following tasks, first change the working directory to ...

```
adapters/bin
```

... in the installation directory.

- **Start:** `./service.sh start`
- **Shut Down:** `./service.sh stop`
- **Restart:** `./service.sh restart`
- **Check Status:** `./service.sh status`

Testing the Web Service on Unix

Perform the following steps to verify that the web service for Netcool/OMNIbus is functional.

1. Open a terminal and change the working directory to the `adapters/bin` directory in the installation directory.
2. Run the test script:

```
./testAdapter.sh
```

3. When the utility prompts for the web service password, enter the password you specified for the web service for Netcool/OMNIBus in step 9 of [Installing the Web Service on Unix](#).

If the test completes successfully, the last line the utility displays is "Test completed successfully."



Note:

Starting with IBM Tivoli Netcool/OMNIBus Connector Release 13.2.1.0.0, the `testAdapter` script is no longer available.

Copying the Required .jar Files

The web service for Netcool/OMNIBus requires you to copy the following .jar files in the Web Services installation directory to the `lib/adapters` directory::

- `log.jar` — Used by the EIF probe
- `evd.jar` — Used by the EIF probe
- `jconn3.jar` — Sybase JDBC driver file

To copy the appropriate .jar files, complete the following steps:

1. From the Object Server system, access the following directory:
`$OMNIHOME/java/jars`
2. Copy the `log.jar`, `jconn3.jar` and `evd.jar` files to the `lib/adapters` directory in the web service installation directory.

 **Note:**

Periodically, new versions of the JDBC file are released. When a major release occurs, the name of the `jconn.jar` file changes. Whenever the jar file name changes, a configuration change must be made to the OMNIBus web service. Listed below are the steps required to make the configuration change to the web service:

1. Navigate to the `conf` directory of the Web Service installation directory.
2. Open the `framework.properties` file in a text editor.
3. Change the `omnibus.sql.driver` property to the path of the SybDriver class in the `jconn` jar file that was downloaded. For example:

If the jar file is named `jconn5.jar`, then the path should be:

```
com.sybase.jdbc5.jdbc.SybDriver
```

If the jar file is named `jconn6.jar`, then the path should be:

```
com.sybase.jdbc6.jdbc.SybDriver
```

If there are any problems using those paths, you can use the `jar` utility to determine the correct path to the SybDriver class as shown below

```
jar tvf /path/to/jconnx.jar | grep SybDriver
```

1. Save the update to the `framework.properties` file.
2. Restart the OMNIBus web service.

Uninstalling the Web Service on Unix

To uninstall the web service on Unix:

1. Run the `service.sh status` command to determine whether the web service is running.
2. If the web service is running, run the `service.sh stop` command to stop the web service and verify it completes successfully.
3. Delete all files in the installation directory.

Web Service for Netcool/OMNIBus on Windows

The following sections explain how to install and then subsequently run the web service for Netcool/OMNIBus on a Windows platform:

- [Prerequisites](#)
- [Installing the Web Service on Windows](#)
- [Running the Web Service on Windows](#)
- [Testing the Web Service on Windows](#)
- [Copying the Required .jar Files](#)
- [Uninstalling the Web Service on Windows](#)

Prerequisites

The following prerequisites must be met before proceeding to the next section.

- IBM Tivoli Netcool/OMNibus Connector Release 13.2.1.0.0 and 13.2.2.0.0 requires JDK 8 Update 5 and later or JDK 11 (any version) installed on the Windows platform.
- IBM Tivoli Netcool/OMNibus Connector Release 12.1.0.x.0 supports only JDK 6 and requires JDK 6 Update 11 and later.
- JAVA_HOME environment variable is set to the JDK installation directory.
- Zip utility, such as WinZip, is installed for unzipping a zip file.



Note:

IBM Tivoli Netcool/OMNibus Connector does not support JDK 7, JDK 9, and JDK 10.

Installing the Web Service on Windows

To install the web service on a Windows platform, perform the following steps:

1. Create a directory where you want to install the web service.
2. Copy the `OMNibus_webservices_adapter.jar` file from the Management Server host system to the web service installation directory. The file is located in the archives directory that was extracted in [Exporting the Installation Files](#)
3. Open a command prompt window and change the working directory to the web service installation directory.
4. Enter the following command to extract the web services components from the `web services .jar` file:

```
jar xvf OMNibus_webservices_adapter.jar
```



Note:

If the system where the web service for Netcool/OMNibus is being installed does not have the JDK installed, you cannot extract the jar file contents. You need to copy the jar file to a system that has the installed JDK and transfer the files after they have been extracted.

This creates the `adapters` directory that contains the installation files.

5. Enter the following command to change the working directory as follows:

```
cd adapters\endpoints\omnibus
```

6. Enter the following command to run the installation script:

```
install.bat
```

7. When the script prompts whether you want to use HTTPS:
 - If you specify **Y**, then the web service is set up to use HTTPS port number 8443.
 - If you specify **N**, then the web service is set up to use HTTP port number 8080.
8. When the script prompts for the user name of the web service, enter a user name that must be provided to access the web service for Netcool/OMNIBus.

The user name can be any value and is not associated with any specific OS or Netcool/OMNIBus account. Note this value and supply it when configuring the Netcool/OMNIBus connector in Enterprise Manager.
9. When the script prompts for the password of the web service, enter the password that must be provided to access the web service for Netcool/OMNIBus.

The password can be any value and is not associated with any specific OS or Netcool/OMNIBus account. Note this value and supply it when configuring the Netcool/OMNIBus connector in Enterprise Manager.
10. When the script prompts for the username to access the Object Server database, enter the username of an account that has read access to the `alerts.status`, `alerts.details`, and `alerts.journals` database tables.
11. When the script prompts for the password to access the Object Server database, enter the password for the account specified in the previous step.
12. When the script prompts for the host name of the Object Server system, enter the host name or IP address of the system where the Object Server is installed. You will not be allowed to specify a host name of localhost, because the web service will fail when attempting to connect to the Object Server database.
13. When the script prompts for the Object Server database port number, enter the port number to use when connecting to the Object Server database. The default port number is **4100**. To determine the port number used by your system, start the Netcool/OMNIBus Administrator and open the Object Servers window. The port number for the Object Server is listed on this window.
14. The web service can be configured to attempt to connect to a backup Object Server database if there are problems connecting to the primary Object Server database. You will be prompted to specify whether you want to configure the web service to connect to a backup Object Server instance. If you answer **N** to the prompt, skip to step 17.
15. When the script prompts for the host name of the backup Object Server system, enter the host name or IP address of the system where the backup Object Server is installed. You will not be allowed to specify a hostname of localhost, because the web service will fail when attempting to connect to the Object Server database.
16. When the script prompts for the backup Object Server database port number, enter the port number to use when connecting to the backup Object Server database. The default port number is **4100**. To determine the port number used by your system, start the Netcool/OMNIBus Administrator and open the Object Servers window. The port number for the Object Server is listed on this window.
17. When the script prompts for the master EIF probe hostname, enter the host name or IP address of the system where the master TEC EIF probe is installed.
18. When the script prompts for the master EIF probe port number, enter the port number to use when connecting to the master TEC EIF probe. The port number is defined in the `PortNumber` property in the `tivoli_eif.props` file. The file is located in the EIF probe installation directory.

19. You can also configure the web service to also send events to a slave EIF probe when sending events to the master EIF probe. You will be prompted to specify whether you want to configure the web service to send events to a slave EIF probe. If you answer **N** to the prompt, skip to step 22.
20. When the script prompts for the slave EIF probe hostname, enter the host name or IP address of the system where the slave TEC EIF probe is installed.
21. When the script prompts for the slave EIF probe port number, enter the port number to use when connecting to the slave TEC EIF probe. The port number is defined in the `PortNumber` property in the `tivoli_eif.props` file. The file is located in the EIF probe installation directory.
22. After the script displays the message "Netcool/OMNIbus Adapter Install Complete," click **Enter** to complete the installation.
23. Copy additional required jar files as specified in [Copying the Required .jar Files](#)
24. If the web service was configured to run using the HTTPS protocol, you must set up SSL as specified in [Configure Oracle Enterprise Manager to Use SSL](#).
25. Delete the `OMNIbus_webservices_adapter.jar` file from the installation directory.

The web service framework is now installed and ready to start.

Running the Web Service as a Windows Service (Optional)

Optionally, if you want the web service to run as a Windows service, perform the following steps:

1. Change the working directory to the `adapters\bin` directory in the installation directory.
2. If the adapter will be run using a 64-bit JRE, copy the `iWaveAdapters.exe` and `iWaveAdaptersw.exe` files from the `x64` directory to the current directory, overwriting the existing files.
3. Enter the following command to install the web service as a Windows service:

```
service.bat install
```

Running the Web Service on Windows

You can run the web service either as a standalone service or Windows service.

Running as a Standalone Service

To start the web service for Netcool/OMNIbus framework when set up as a standalone application (not set up to run as a Windows service):

1. Change the working directory to the `adapters\bin` directory in the installation directory.
2. Run the following command:

```
startAdapters.bat
```

To shut down the web service Netcool/OMNIbus framework, close the window where you started the web service.

Running as a Windows Service

To start the web service for Netcool/OMNIBus framework when set up to run as a Windows service, enter the following command:

```
net start iWaveAdapters
```

To shut down the web service Netcool/OMNIBus framework, enter the following command:

```
net stop iWaveAdapters
```

Testing the Web Service on Windows

Perform the following steps to verify that the web service for Netcool/OMNIBus is functional.

1. Open a terminal and change the working directory to the `adapters\bin` directory in the installation directory.

2. Enter the following command to run the test script:

```
.\testAdapter.bat
```

3. When the utility prompts for the web service password, enter the password you specified for the web service for Netcool/OMNIBus in step 9 of [Installing the Web Service on Windows](#).

If the test completes successfully, the last line the utility displays is "Test completed successfully."



Note:

Starting with IBM Tivoli Netcool/OMNIBus Connector Release 13.2.1.0.0, the `testAdapter` script is no longer available.

Copying the Required .jar Files

The web service for Netcool/OMNIBus requires you to copy the following .jar files in the Web Services installation directory to the `lib/adapters` directory:

- `log.jar` — Used by the EIF probe
- `evd.jar` — Used by the EIF probe
- `jconn3.jar` — Sybase JDBC driver file

To copy the appropriate .jar files, complete the following steps:

1. From the Object Server system, access the following directory:

```
%OMNIHOME%\java\jars
```

2. Copy the `log.jar`, `jconn3.jar` and `evd.jar` files to the `lib/adapters` directory in the web service installation directory.

 **Note:**

Periodically, new versions of the JDBC file are released. When a major release occurs, the name of the `jconn.jar` file changes.

Whenever the jar file name changes, a configuration change must be made to the OMNibus web service. Listed below are the steps required to make the configuration change to the web service:

1. Navigate to the `conf` directory of the web service installation directory.
2. Open the `framework.properties` file in a text editor.
3. Change the `omnibus.sql.driver` property to the path of the `SybDriver` class in the `jconn.jar` file that was downloaded. For example:

If the jar file is named `jconn3.jar`, then the path should be:

```
com.sybase.jdbc3.jdbc.SybDriver
```

If the jar file is named `jconn4.jar`, then the path should be:

```
com.sybase.jdbc4.jdbc.SybDriver
```

1. Save the update to the `framework.properties` file.
2. Restart the OMNibus web service.

Uninstalling the Web Service on Windows

To uninstall the web service on Windows:

1. Determine if the web service is installed as a Windows service:
 - If the web service is installed as a Windows service, perform the following steps:
 - a. Determine if the web service is running.
 - b. If the web service is running, run the `service.bat stop` command to stop the web service and verify it completes successfully.
 - c. Run the `service.bat uninstall` command to remove it as a Windows service and verify it completes successfully.
 - If the web service is not installed as a Windows service, perform the following steps:
 - a. Determine if the web service is running.
 - b. If the web service is running, stop the web service by closing the Java window.
2. Delete all files in the installation directory.

Installing the Oracle Agent for Netcool/OMNIBus

Starting with IBM Tivoli Netcool/OMNIBus Connector Release 13.2.1.0.0, there is no longer a requirement to install the Oracle Agent. This section only applies for IBM Tivoli Netcool/OMNIBus Connector Release 12.1.0.x.0.

If you are upgrading the connector from 12.1.0.x.0 to 13.2.x.0.0, see [Uninstalling the Oracle Agent for Netcool/OMNIBus](#).

The Oracle Agent for Netcool/OMNIBus is invoked by an Object Server external procedure to send event information to the web service for Netcool/OMNIBus.



Note:

The IBM Tivoli Netcool/OMNIBus Process Control/Process Agent must first be installed on the system where the Agent is to be installed. Oracle recommends installing the Oracle Agent for Netcool/OMNIBus on the Object Server system.

To install the Oracle Agent for Netcool/OMNIBus, perform the following steps:

1. Create a directory where you want to install the Oracle Agent for Netcool/OMNIBus.
2. Determine the name of the installation file based on the OS platform where the agent is being installed.

The platforms and associated installation files are as follows:

- Windows — OMNIBusAgentWindows.zip
 - AIX — OMNIBusAgentAIX.tar.gz
 - Linux — OMNIBusAgentLinux.tar.gz
 - Oracle Solaris — OMNIBusAgentSolaris.tar.gz
3. Transfer the appropriate installation file from the Management Server host system to the Agent installation directory. The file is located in the archives directory that was extracted in [Exporting the Installation Files](#).
 4. Extract the files from the zip archive. This creates the agent directory that contains the installation files.
 5. Change the working directory to the agent directory.
 6. Run the setup script to configure the agent.

The file name depends on the platform. If the platform is Windows, the setup script is named `Setup.cmd`. For all other platforms, the setup script is named `setup.sh` and should be called using the following command:

```
sh ./setup.sh
```

7. When the script prompts for the web service host name, enter the host name or IP address where the web service for Netcool/OMNIBus is installed.
8. When the script prompts for the web service port, enter the port number the web service for Netcool/OMNIBus uses to receive transactions from the Agent.

The default port number of **8080** will be used most of the time and will even be used if the adapter is configured to use HTTPS. The only time the default port number will not be used is when the default port is changed as specified in the [Changing the Web Service Port \(8080\)](#) section in [Web Service for Netcool/OMNIBus Details](#). If you are unsure of what port number to use, perform the following steps to determine the port number to use:

- a. At the system where the web service is installed, navigate to the `adapters/conf` directory.
 - b. Open the `framework.properties` file in a text editor and search for the `notification.url` property.
 - c. Whatever port number is specified in the `notification.url` property is the port number you should specify.
9. When the script prompts for the user ID to specify when creating event journal entries in Netcool/OMNIBus, enter the user ID of a user that has permission to add journal entries. If no value is specified, the user ID will default to `0` (`root`). The value specified must be an integer value and will not work properly if a different value is entered.

The script generates the necessary configuration files and places them in the Agent directory. The Agent is now installed, but you need to make some modifications in Netcool/OMNIBus for everything to function properly.

10. Perform the steps in [Modifying Netcool/OMNIBus](#) to complete the setup.

Uninstalling the Oracle Agent for Netcool/OMNIBus

To uninstall the Oracle Agent for Netcool/OMNIBus:

1. If the agent is being permanently deleted, the OMNIBus server modifications must be backed out as specified in [Uninstalling Modifications to Netcool/OMNIBus](#).
2. Delete all files in the installation directory.

Note:

If the Oracle Agent for Netcool/OMNIBus is being replaced, the database external procedures that call the agent will fail until the replacement agent has been installed in the same location.

Modifying Netcool/OMNIBus

The procedures in the following sections enable you to configure Netcool/OMNIBus to work with the installed Oracle Agent and web service components:

- [Configuring the Object Server Database](#) to create a custom database table and the triggers/procedures the Oracle Agent and web service require.
- [Configuring the TEC EIF Probe](#) to change the configuration of the TEC EIF probe to populate the custom database table when creating/updating events in Netcool/OMNIBus.

- [Change Deduplication Trigger Priority](#) to change the configuration of the TEC EIF probe to populate the custom database table when creating/updating events in Netcool/OMNIBus.

Configuring the Object Server Database

Perform the following steps to run the configuration script that sets up the Netcool/OMNIBus Object Server database to work with the installed Oracle Agent and web service components.

1. If the Oracle Agent for Netcool/OMNIBus was NOT installed on the primary Object Server system, copy the `configure_oracle_webservice.sql` file from the system where the Oracle Agent for Netcool/OMNIBus was installed to the system where the primary Object Server is installed.
2. Run the SQL script using the appropriate command listed below based on the applicable platform:

Unix:

```
$OMNIBUSHOME/bin/nco_sql -S <objserver> -U <username> -P <password>  
< <script_directory>/configure_oracle_webservice.sql
```

Windows:

```
"%OMNIBUSHOME%\bin\isql" -S <objserver> -U <username> -P <password>  
-i <script_directory>\configure_oracle_webservice.sql
```

When entering the commands, provide the following information:

- `objserver` — Object Server name
- `username` — User name that will be used when connecting to the database. The account must have permission to create tables, triggers, and procedures in the Object Server database.
- `password` — Password associated with the user name
- `script_directory` — Directory where the `configure_oracle_webservice.sql` file is located. If the Oracle Agent for Netcool/OMNIBus was installed on the primary Object Server system, this will be the Oracle Agent for Netcool/OMNIBus installation directory. Otherwise, it will be the directory where the file was copied.

Configuring the TEC EIF Probe

You need to manually change the set up of the TEC EIF probe configuration to accept alerts from the web service for Netcool/OMNIBus. You must modify the existing EIF probe rules file to include an Oracle-supplied rules file.



Note:

If the probes are configured to operate in a master/slave pair, these changes are required for each probe.

To install the new rules file, perform the following steps:

1. Shut down the EIF probe.

2. Access the EIF probe installation directory.
3. Make a back-up copy of the existing `tivoli_elf.rules` file.
4. Copy the `oracle_elf.rules` file from the system where the Oracle Agent for Netcool/OMNIBus was installed to the EIF probe install directory.
5. Modify the `tivoli_elf.rules` file as follows:
 - a. Insert the following statements before any other executable statements:

```
defaultAlerts = registertarget("<PRIMARY>", "<BACKUP>", "alerts.status")
OracleAlerts = registertarget("<PRIMARY>", "<BACKUP>",
"custom.oracle_status")
```
 - b. Replace `<PRIMARY>` with the name of the primary object server.
 - c. Replace `<BACKUP>` with the name of the back-up server. If a back-up is not configured, replace it with an empty string so that the back-up server is `""`.
 - d. In the location where the `tivoli_elf_tpc.rules` and `tivoli_elf_tsm.rules` files are included, add an include statement for the `oracle_elf.rules` file.
 - e. Save the `tivoli_elf.rules` file.
6. Make sure the Inactivity parameter for the `tivoli_elf.props` file is set to **0**.
7. Restart the EIF probe.

 **Note:**

After installing the EIF Probe, at a minimum, you need to edit the `tivoli_elf.props` file and set the correct Netcool/OMNIBus server name. The file can be found in the following locations:

For UNIX:

```
%OMNIHOME%/probes/<platform>/tivoli_elf.prop
```

For Windows:

```
%OMNIHOME%\probes\<platform>\tivoli_elf.prop
```

Where:

`<platform>` is the actual platform name (such as `win32`).

Refer to the *IBM Netcool/OMNIBus Installation and Deployment Guide* for more information on configuring the EIF Probe:

http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/index.jsp?topic=%2Fcom.ibm.tivoli.namomnibus.doc%2Fwelcome_ob.htm

Change Deduplication Trigger Priority

The `oracle_alert_reinserted` trigger in the `primary_only` group must run first in order to function properly. To ensure the trigger runs first, the priority of the deduplication trigger in the `default_triggers` group must be changed from **1** to **2**. The trigger priority can be modified using the Netcool/OMNIBus Administrator application.

**Note:**

On some older versions of Netcool/OMNIBus, this trigger does not exist. If the trigger does not exist in your system, then no changes are required in this section.

Uninstalling Modifications to Netcool/OMNIBus

To uninstall modification made to Netcool/OMNIBus:

Revert Deduplication Trigger Priority

Change the priority of the deduplication trigger in the `default_triggers` group back to **1**. The trigger priority can be modified using the Netcool/OMNIBus Administrator application.

Revert TEC EIF Probe Configuration

Perform the following to revert the TEC EIF probe configuration back to its original settings:

1. Shut down the EIF probe.
2. Access the EIF probe installation directory.
3. Replace the `tivoli_eif.rules` file with the backup copy of the file that was created during installation.
4. Delete the `oracle_eif.rules` file from the EIF probe install directory.
5. Restart the EIF probe.

Revert the Object Server Database Configuration

Perform the following steps to uninstall the customizations made to the Netcool/OMNIBus Object Server database during installation:

1. If the Oracle Agent for Netcool/OMNIBus was NOT installed on the primary Object Server system, copy the `uninstall_oracle_webservice.sql` file from the system where the Oracle Agent for Netcool/OMNIBus was installed to the system where the primary Object Server is installed.
2. Run the SQL script using the appropriate command listed below based on the applicable platform:

- Unix:

```
$OMNIBUSHOME/bin/nco_sql -S <objserver> -U <username> -P <password>  
<script_directory>/uninstall_oracle_webservice.sql
```

- Windows:

```
"%OMNIBUSHOME%\bin\isql" -S <objserver> -U <username> -P <password> -i  
<script_directory>\uninstall_oracle_webservice.sql
```

When entering the commands, provide the following information:

- `objserver` — Object Server name
- `username` — User name that will be used when connecting to the database. The account must have permission to delete tables, triggers, and procedures from the Object Server database.

- `password` — Password associated with the user name
- `script_directory` — Directory where the `uninstall_oracle_webservice.sql` file is located. If the Oracle Agent for Netcool/OMNIbus was installed on the primary Object Server system, this will be the Oracle Agent for Netcool/OMNIbus installation directory. Otherwise it will be the directory where the file was copied.

Uninstalling the Connector

To uninstall the connector, you must first delete all defined instances of the connector, then you must delete the connector from the Self Update page.

1. From the Setup menu, select **Extensibility**, then **Management Connectors**.
2. Select an instance of the connector you want to delete, then click **Delete**.
3. On the Confirmation page, click **Yes**.
4. Repeat steps 2 and 3 until all instances of the connector have been deleted.
5. From the Setup menu, select **Extensibility**, then **Self Update**.
6. Click on the **Management Connector** link in the Type column. A list of updates appears for Management Connectors.
7. Click on the connector you want to delete, select **Actions**, then select **Delete**. The Delete Update window appears.
8. Click **Delete** to delete the connector. A pop-up confirmation window appears.
9. Click **OK** to confirm and delete the connector.

3

Configuring the IBM Tivoli Netcool/OMNIBus Connector

This chapter provides procedures to configure the two sub-pages of the main Configure Management Connector page, then explains how to perform other tasks to complete the configuration process.

This chapter covers the following topics:

- [Creating a Connector Instance](#)
- [Configuring the Connector](#)
- [Enabling and Disabling the Connector](#)
- [Providing General Settings](#)
- [Testing the IBM Tivoli Netcool/OMNIBus Connector](#)

Creating a Connector Instance

After the IBM Tivoli Netcool/OMNIBus Connector is installed in Oracle Enterprise Manager, you need to create one or more instances of this connector type. A connector instance refers to the configured instance of the connector with specific details, such as connector name, web-service end-points, username/password, and so forth of the Netcool/OMNIBus system. Multiple connector instances for the same connector type are supported, which provides the flexibility to configure multiple Netcool/OMNIBus systems with Enterprise Manager.

For example, Enterprise Manager is monitoring both test and production targets. You could create two different connector instances that point to two different Netcool/OMNIBus systems such that events from production targets are forwarded to one Netcool/OMNIBus system and events from test targets are forwarded to the second Netcool/OMNIBus system.

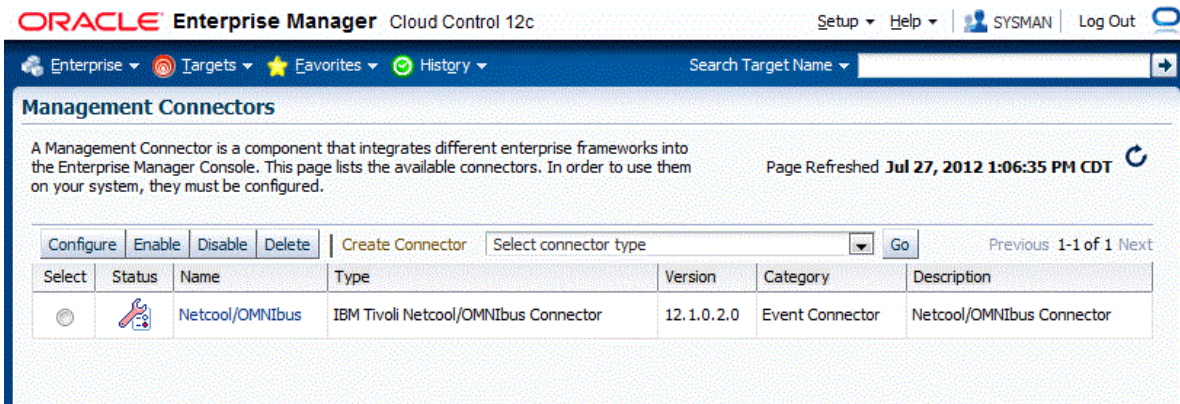
To create a connector instance:

1. From the **Setup** menu of Enterprise Manager Cloud Control, select **Extensibility**, then **Management Connectors**.
2. Select a connector type of **IBM Tivoli Netcool/OMNIBus Connector 12.1.0.2.0** from the Create Connector drop-down list, then click **Go**.
3. Enter the connector name and description on the next page and click **OK**.

The connector instance appears as unconfigured in the Connector Setup page.

[Figure 3-1](#) shows an unconfigured IBM Tivoli Netcool/OMNIBus connector instance.

Figure 3-1 Unconfigured IBM Netcool/OMNIBus Connector



Configuring the Connector

To configure the connector:

1. From the **Setup** menu of Enterprise Manager Cloud Control as Super Administrator, select **Extensibility**, then **Management Connectors**.

The Management Connectors page appears, which lists all created connector instances.

Note:

The connector is already configured if a green checkmark appears in the Status column.

2. Click on the name of the Netcool/OMNIBus Connector.

The General tab of the Configure Management Connector page appears, as shown in [Figure 3-2](#).

Figure 3-2 Configure Management Connector Page

The screenshot shows the 'Configure Management Connector: Netcool/OMNIBus' page in Oracle Enterprise Manager. The page has tabs for 'General', 'Template', and 'Targets'. The 'General' tab is active. A note states: 'In order to configure this connector, the pre-requisite steps must already have been completed on the external system.' The 'Connection Settings' section includes a table for 'Web Service End Points' with columns for 'Operation' and 'Web Service End Point (URL)'. The table contains two rows: 'createEvent' with URL 'https://{Hostname}:8443/services/omnibus/EventService' and 'updateEvent' with the same URL. Below the table are input fields for 'Netcool/OMNIBus Web Service Username' and 'Netcool/OMNIBus Web Service Password'. The 'Retry' section has a checkbox for 'Enable Retry' (which is unchecked) and an 'Expiration (Hours)' field set to '0'. 'Cancel' and 'OK' buttons are present at the top right and bottom right of the configuration area.

3. Provide the required settings. See [Providing General Settings](#) for details.
4. Click **OK**.

The Management Connectors page reappears. The row for the Netcool/OMNIBus Connector should have a check mark in the Configured column.

 **Note:**

If you chose HTTPS as the protocol to establish a connection between Netcool/OMNIBus and Enterprise Manager, make sure you have performed the steps specified in [Configure Oracle Enterprise Manager to Use SSL](#).

Enabling and Disabling the Connector

You can enable or disable a connector instance without needing to reconfigure or remove the connector from the system.

You can enable or disable a connector from the Management Connector page by selecting the connector and clicking either the **Enable** or **Disable** button. After a connector is disabled, the **X** icon appears under the Status column. By default, the connector instance is enabled as soon as it is configured.

 **Note:**

Only enabled connectors are available to forward events.

Providing General Settings

The following sections explain how to provide various configuration details:

- [Connection Settings](#)
- [Retry](#)

Connection Settings

The IBM Tivoli Netcool/OMNIBus Connector communicates with Netcool/OMNIBus through the web service for Netcool/OMNIBus installed in [Installing the Web Service for Netcool/OMNIBus Front-End](#). All of the following fields are mandatory:

- **Web Service End Points:** URL to use when creating and updating alerts in Netcool/OMNIBus. The web service for Netcool/OMNIBus uses the same URL for the `createEvent` and `updateEvent` operations. The default format for both operations is as follows:

```
https://[Hostname]:8443/services/omnibus/EventService
```

Modify the URL for both operations based on whether the web service for Netcool/OMNIBus was set up using HTTP or HTTPS.

For HTTP:

- Change the protocol from `https` to `http`.
- Replace `[Hostname]` in the URL with the host name or IP address of the system where the web service for Netcool/OMNIBus was installed.
- Change the port number from `8443` to `8080`.

For HTTPS:

- Replace `[Hostname]` in the URL with the hostname or IP address of the system where the web service for Netcool/OMNIBus was installed.

For example:

If the web service for Netcool/OMNIBus was installed on a system with a host name of `omnibusws`, the following URLs would be configured for the different protocols:

- For HTTP: `http://omnibusws:8080/services/omnibus/EventService`
- For HTTPS: `https://omnibusws:8443/services/omnibus/EventService`
- **Netcool/OMNIBus Web Service Username:** User name that was configured for accessing the web service for Netcool/OMNIBus in step 8 of the appropriate web service installation section:
 - [Installing the Web Service on Unix](#).
 - [Installing the Web Service on Windows](#).
- **Netcool/OMNIBus Web Service Password:** Password that was configured for accessing the web service for Netcool/OMNIBus in step 9 of the appropriate web service installation section:
 - [Installing the Web Service on Unix](#).

- [Installing the Web Service on Windows.](#)

Retry

The Retry section, available on the connector configuration page, enables you to specify whether failed operations need to be retried within a configurable expiration time.

When you enable the Retry option, you can specify whether you want to resend a Create or Update request if the request fails the first time, and specify the time period after which you want to abandon the retry. Enterprise Manager retries every five minutes until the request is successful or the retry interval expires.

By default, the Retry is disabled. This setting applies to all events the Netcool/OMNIBus Connector processes. The retry period interval accepts the number of hours as any integer value; for example, **1** or **2**.

Testing the IBM Tivoli Netcool/OMNIBus Connector

To verify that the connector is configured correctly:

1. Log in to Oracle Enterprise Manager Cloud Control with an account that has Super Administrator permissions.
2. From the **Setup** menu, select **Extensibility**, then **Management Connectors**.
The Management Connectors page appears.
3. Click on the name of the Netcool/OMNIBus Connector.
The General tab of the Configure Management Connector page appears.
4. Select and copy the URL specified for the `createEvent` or `updateEvent` operation.
5. Open an internet browser on the system where the Oracle Enterprise Manager server is installed.
6. In the address window, enter the URL that was copied in step 4 above. Add `?wsdl` to the end of the URL. The URL should appear similar to the following example:

```
https://[Hostname]:8443/services/omnibus/EventService?wsdl
```

[Hostname] is the actual host name or IP address where the web service for Netcool/OMNIBus is installed.

If the WSDL is loaded, this confirms that the connector is configured correctly for sending event information to Netcool/OMNIBus.

4

Forwarding Events to IBM Netcool/OMNIBus

Events generated or updated in Oracle Enterprise Manager are not transferred to Netcool/OMNIBus unless you create rules to invoke the IBM Tivoli Netcool/OMNIBus Connector. A rule identifies the conditions that must be met before the connector is invoked.

The following sections provide procedures that explain how to create and update rules:

- [Setting Event Rules](#)
- [Viewing Events in the IBM Netcool/OMNIBus Console](#)

Setting Event Rules

This section explains how to create new rules in Enterprise Manager that invoke the Netcool/OMNIBus connector.

To set events rules:

1. Log in to Enterprise Manager Cloud Control with a user account that has the Create Enterprise Rule Set resource privilege and at least View privileges on the targets where events are to be forward to Netcool/OMNIBus.
2. From the **Setup** menu, click **Incidents**, then **Incident Rules**.
3. Identify an existing rule set where the new rule can be added, or create a new rule set as follows:
 - a. Click **Create Rule Set** to invoke the Create Rule Set page.
 - b. Enter a name and description, then click **Save**.

The Incident Rules page appears again, and lists the new rule set.

4. Click the rule set for which you want to add the rule, then click **Edit**.
5. Click the **Rules** tab then click **Create**.

The Select Type of Rule to Create pop-up appears. As you proceed below, the wizard interface explains various rules and their usage. For detailed information, see the "Incident Management" chapter in the *Oracle Enterprise Manager Cloud Control Administrator's Guide*.

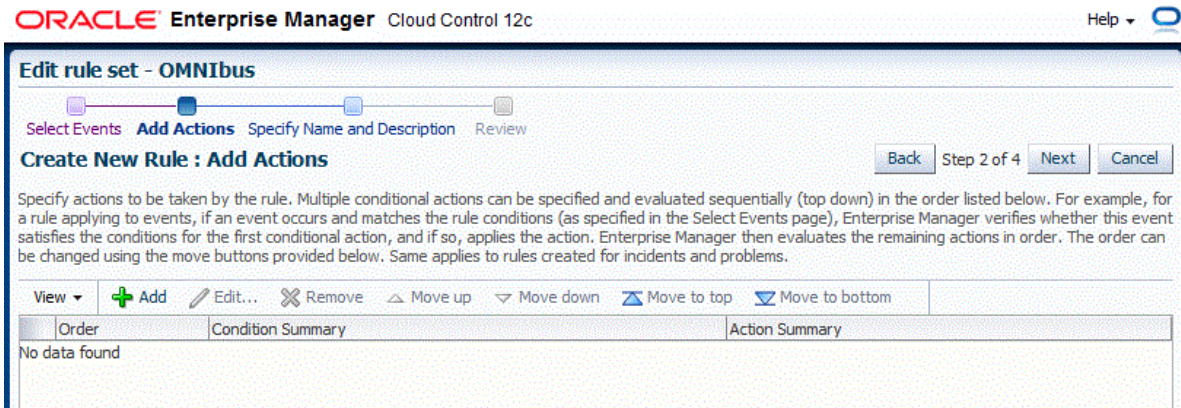
6. Select **Incoming events and updates to events**, then click **Continue**.

The Select Events page appears.

7. Select the criteria for events that you want to forward to Netcool/OMNIBus. If any new event matches the criteria specified, then the rule will be invoked to send the event to Netcool/OMNIBus. Click **Next**.

The Add Actions page will be displayed as shown in [Figure 4-1](#).

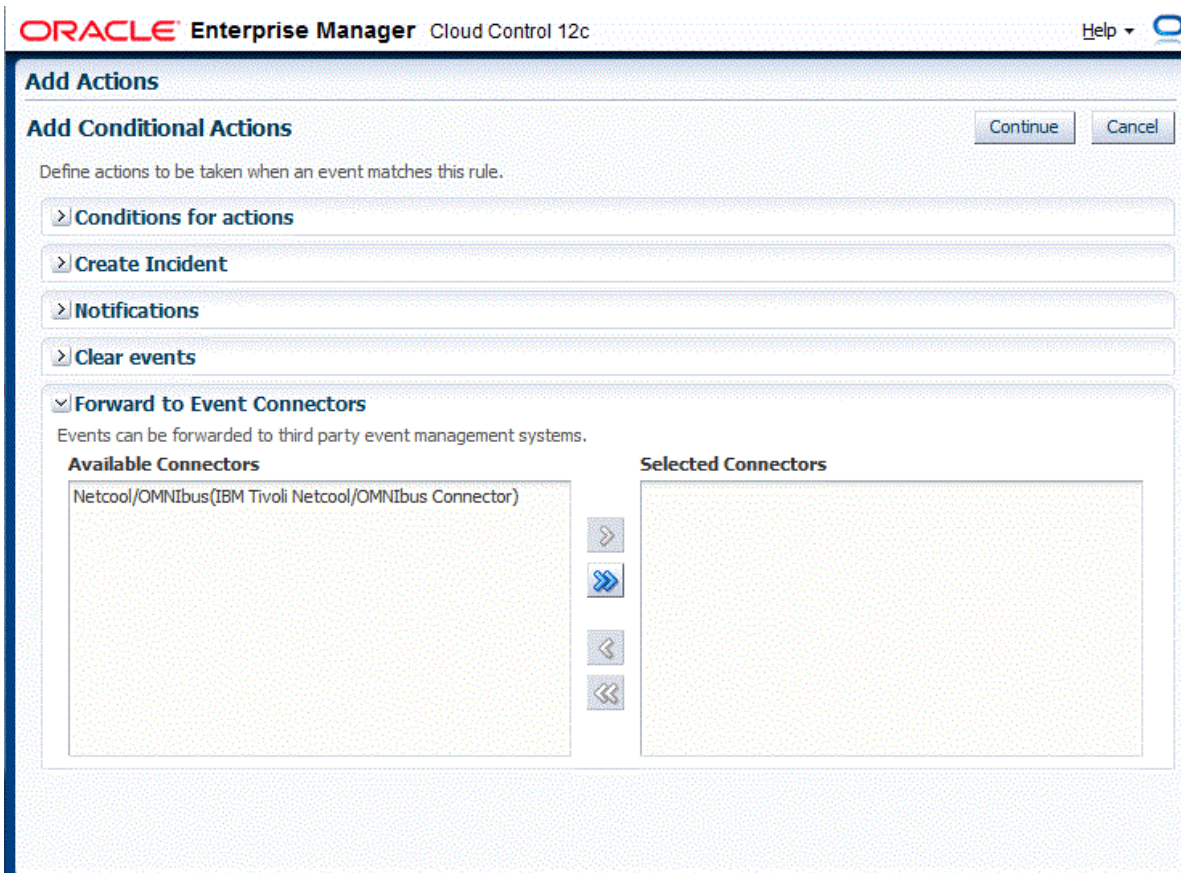
Figure 4-1 Add Actions Page



8. Click **Add**.

The Add Conditional Actions page appears, as shown in [Figure 4-2](#).

Figure 4-2 Add Conditional Actions Page



9. Select one or more connector instances listed in the Forward to Event Connectors section and, click the > button, then click **Continue**.

The Add Actions page appears again, and lists the new action.

10. Click **Next**.

The Specify Name and Description page appears.

11. Enter a name and description for the rule, then click **Next**.

The Review page appears.

12. Click **Continue** if everything appears correct.

An information pop-up appears that states, "Rule has been successfully added to the current rule set. Newly added rules are not saved until the Save button is clicked."

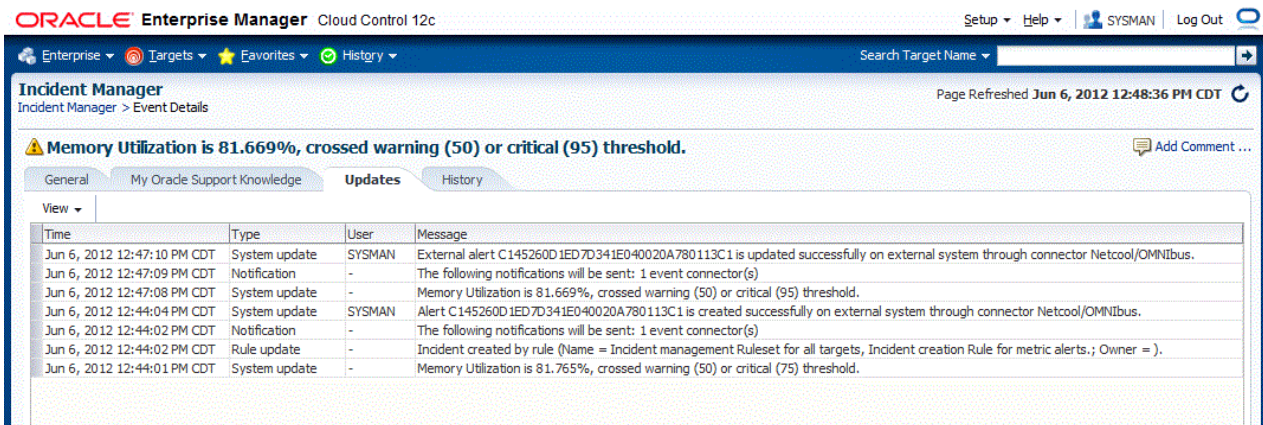
You can click **Back** and make corrections to the rule if necessary.

13. Click **Save** to save the changes to the rule set.

The following process occurs after the event rule has been created successfully:

1. A request is sent to the Netcool/OMNIbus Connector when an event is created per the criteria mentioned in the rule.
2. The alert is created or updated on the Netcool/OMNIbus console.
3. In Enterprise Manager, the event is annotated with a comment on the Updates tab to indicate that an event is created or updated, as shown in [Figure 4-3](#).

Figure 4-3 Event Annotations

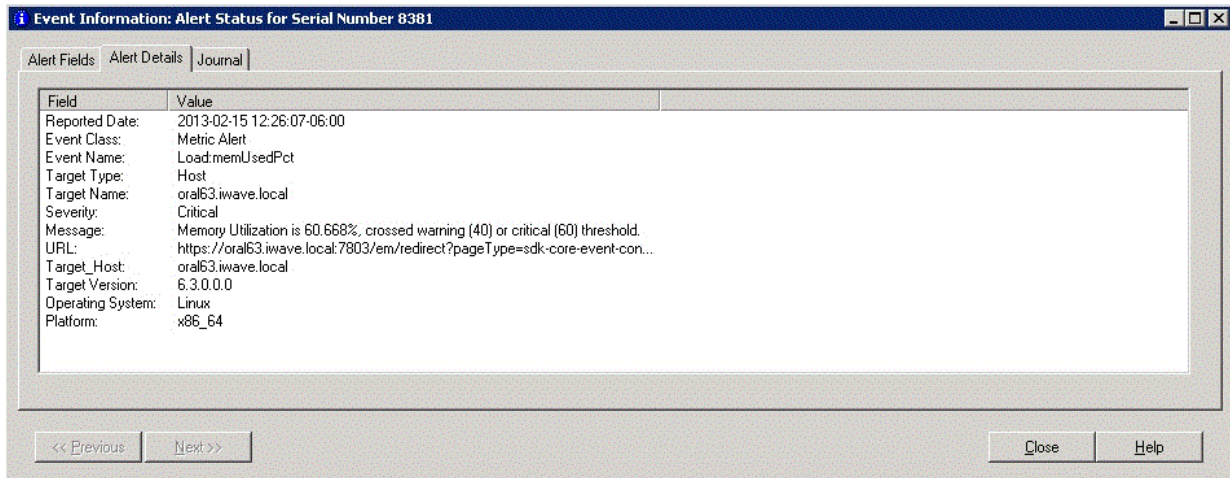


Viewing Events in the IBM Netcool/OMNIbus Console

After the event has been sent to IBM Netcool/OMNIbus, you can view it in the IBM Netcool/OMNIbus console, as shown in [Figure 4-4](#). Full details of the event are sent to IBM Netcool/OMNIbus, including:

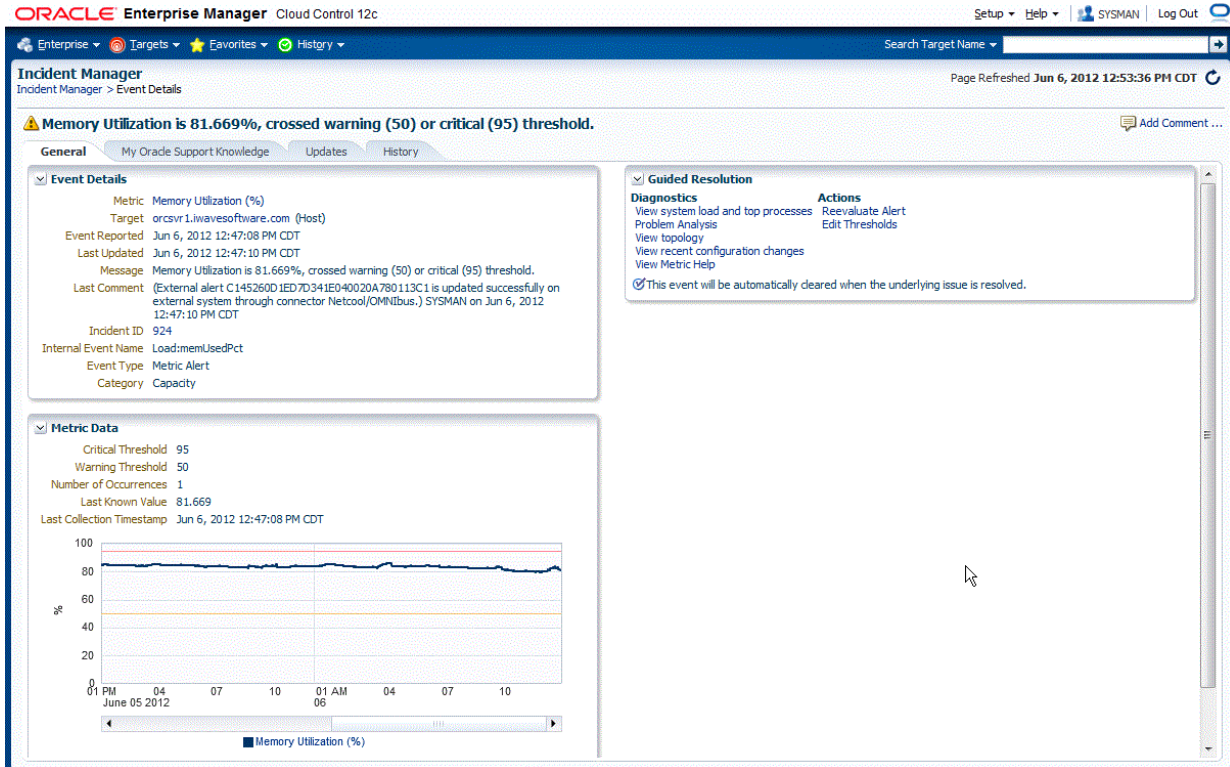
- Event class and name.
- Target type and name.
- Message.
- Severity.
- URL to the Enterprise Manager event details page.

Figure 4-4 Event Details in the IBM Netcool/OMNIBus Console



To view the event in Enterprise Manager, copy the URL listed in the event and paste in the address bar of a browser. You will be asked to log in to Oracle Enterprise Manager. After logging in, the Oracle Enterprise Manager event information is displayed, as shown in Figure 4-5.

Figure 4-5 Event Details in Oracle Enterprise Manager



You can change the event information sent to IBM Netcool/OMNIBus, or you can change the mappings between the Enterprise Manager event and IBM Netcool/OMNIBus event by modifying the event templates.

See [Working with Event Templates](#) , for details.

5

Working with Event Templates

This chapter provides information about the default event templates shipped with the IBM Tivoli Netcool/OMNIbus Connector.

Event templates are XML transformation style sheets that map Enterprise Manager event attributes to Netcool/OMNIbus alert attributes.

The Netcool/OMNIbus event connector contains predefined event templates for Create and Update event operations on the Netcool/OMNIbus system. You can customize these templates to fulfill your particular requirements. Oracle recommends that you back up these factory-built templates before customizing them.

This chapter discusses the following topics:

- [Default Templates](#)
- [Severity Mappings](#)
- [Customizing Event Mappings](#)

Default Templates

This section describes the default mappings between the Enterprise Manager event data fields and the Netcool/OMNIbus alert data fields.

XML Style Sheet (XSL) files contain the mappings between the two systems. These files are located in the Self Update archive that was installed in [Installing the Connector in Enterprise Manager](#).

To obtain copies of the XSL files, perform the steps in [Exporting the Template Files](#).

[Table 5-1](#) lists the XSL files that perform the mappings and provides a summary of each.

Table 5-1 XSL Files That Perform Mappings

File	Description
<code>createEvent_request.xml</code>	Transforms the Oracle Enterprise Manager event data to the IBM Netcool/OMNIbus alert format for the createEvent operation.
<code>updateEvent_request.xml</code>	Transforms the Oracle Enterprise Manager event data to the IBM Netcool/OMNIbus alert format for the updateEvent operation.

The following sections provide details about the default mappings in each of the files:

- [Create Event Template](#)
- [Update Event Template](#)
- [Details Mapping - Create](#)
- [Details Mapping - Update](#)

Create Event Template

The Oracle Enterprise Manager Connector Framework invokes the `createEvent` operation whenever an event is generated in Oracle Enterprise Manager and there is a rule configured to invoke the Netcool/OMNIBus connector. `createEvent_request.xml` is invoked during the process to transform the data from Oracle Enterprise Manager format to Netcool/OMNIBus alert format. [Table 5-2](#) lists the default field mappings between the IBM Tivoli Netcool/OMNIBus alert and the Oracle Enterprise Manager event.

Table 5-2 Netcool/OMNIBus Field Name

Netcool/OMNIBus Field Name	Value
Identifier	Set to the event <code>EventId</code> field.
Node	Set to the event <code>TargetName</code> field.
NodeAlias	Set to the event <code>TargetName</code> field followed by the <code>Target_Host</code> property information.
Manager	Hard coded to "Oracle Enterprise Manager."
Agent	Set to the event <code>NotificationRuleName</code> field.
Summary	Set to the event <code>Message</code> field.
Severity	Set the severity based on the event severity as specified in Severity Mappings .
AlertGroup	Set to the event <code>EventCategory</code> field.
AlertKey	Set to the <code>EventClass</code> field followed by the event <code>EventName</code> field.
LastOccurrence	Set to the event <code>ReportedDate</code> field.
URL	Set to the event <code>EventURL</code> field.
Details	Set to the formatted contents of the event as specified in Details Mapping - Create .
Type	Problem
Journal	Set to the formatted contents of the event as specified in Journal Mapping .

Update Event Template

The Oracle Enterprise Manager Connector Framework invokes the `updateEvent` operation whenever an event is updated in Oracle Enterprise Manager and there is a rule configured to invoke the Netcool/OMNIBus connector. `updateEvent_request.xml` is invoked during the process to transform the data from Oracle Enterprise Manager format to Netcool/OMNIBus alert format. [Table 5-3](#) lists the default field mappings between the IBM Netcool/OMNIBus alert and the Oracle Enterprise Manager event.

Table 5-3 Update Event Template Mapping

Netcool/OMNIBus Field Name	Value
Identifier	Set to the event ExternalEventID field.
Severity	Set the severity based on the event severity as specified in Severity Mappings .
Summary	Set to the event Message field.
LastOccurrence	Set to the event ReportedDate field.
Details	Set to the formatted contents of the event as specified in Details Mapping - Update .
Journal	Set to the formatted contents of the event as specified in Journal Mapping .

Details Mapping - Create

This section identifies the format of the information that will be placed in the details table whenever an alert is created in Netcool/OMNIBus. The format the template uses for the details table is shown below. Text delimited by angle brackets (< and >) designates an Enterprise Manager Event field name that will be replaced by the actual field contents. Lines marked with double-asterisks indicate they will only be present if the designated event field contains data. The Notes section below the format contains information about some of the fields.

```

Occurred Date: <OccurredDate>
ReportedDate: <ReportedDate>
Event Class: <EventClass>
Event Name: <EventName>
Target Type: <SourceInfo/TargetInfo/TargetTypeLabel>
Target Name: <SourceInfo/TargetInfo/TargetName>
Severity: <Severity>
Message: <Message>
URL: <EventURL>
<TargetProperty/Name[1]>: <TargetProperty/Value[1]>**
<TargetProperty/Name[2]>: <TargetProperty/Value[2]>**
<TargetProperty/Name[3]>: <TargetProperty/Value[3]>**
<TargetProperty/Name[4]>: <TargetProperty/Value[4]>**
<TargetProperty/Name[5]>: <TargetProperty/Value[5]>**
<TargetProperty/Name[6]>: <TargetProperty/Value[6]>**
<TargetProperty/Name[7]>: <TargetProperty/Value[7]>**
<TargetProperty/Name[8]>: <TargetProperty/Value[8]>**
<TargetProperty/Name[9]>: <TargetProperty/Value[9]>**
<TargetProperty/Name[10]>: <TargetProperty/Value[10]>**
<TargetProperty/Name[11]>: <TargetProperty/Value[11]>**
<TargetProperty/Name[12]>: <TargetProperty/Value[12]>**
<TargetProperty/Name[13]>: <TargetProperty/Value[13]>**
<TargetProperty/Name[14]>: <TargetProperty/Value[14]>**
<TargetProperty/Name[15]>: <TargetProperty/Value[15]>**
<TargetProperty/Name[16]>: <TargetProperty/Value[16]>**
<TargetProperty/Name[17]>: <TargetProperty/Value[17]>**
<TargetProperty/Name[18]>: <TargetProperty/Value[18]>**
<TargetProperty/Name[19]>: <TargetProperty/Value[19]>**
<TargetProperty/Name[20]>: <TargetProperty/Value[20]>**

```

Notes

1. The Target Properties information is only present if the event contains target property values. Listed below are a couple of sample target property name/value pairs:

```
Line of business: Finance
Owner: Finance-DB Team
```

2. URL is set to the URL of the Event Details page in Enterprise Manager. The URL must be copied into a browser to view the event in Enterprise Manager.

Details Mapping - Update

This section identifies the format of the information that will be updated in the details table whenever an alert is updated in Netcool/OMNIBus. Text delimited by angle brackets (< and >) designates an Enterprise Manager Event field name that will be replaced by the actual field contents.

```
Severity: <Severity>
Message: <Message>
```

Journal Mapping

This section identifies the format of journal entries that are added whenever an alert is created or update in Netcool/OMNIBus. Text delimited by angle brackets (< and >) designates an Enterprise Manager Event field name that will be replaced by the actual field contents.

```
Severity: <SeverityCode>
Reported Date: <ReportedDate>
<Message>
```

Severity Mappings

The Netcool/OMNIBus alert severity is set based on the Enterprise Manager event severity value. [Table 5-4](#) shows the values used when creating the alert in Netcool/OMNIBus.

Table 5-4 Netcool/OMNIBus Alert Severity Mappings

When the event severity is ...	Set the alert severity to ...
Clear	0
Informational	1
Minor Warning	2
Warning	2
Critical	5
Fatal	5

Customizing Event Mappings

Although the default mappings are sufficient for most implementations, you can change them as needed. The following sections discuss:

- [XML Format of Oracle Enterprise Manager Events](#)
- [Managing Templates in Oracle Enterprise Manager 12c](#) (recommended)
- [Manually Changing a Mapping](#)

It is assumed that you already have a good understanding of XSL.

For reference information on the Netcool/OMNIbus mappings, see [Web Service for Netcool/OMNIbus Details](#).

XML Format of Oracle Enterprise Manager Events

[Example 5-1](#) shows the format that the Oracle Enterprise Manager Connector Framework provides when an event is created or updated in Oracle Enterprise Manager.

For more information about Oracle Enterprise Manager events, see the *Using Incident Management* chapter of the *Oracle Enterprise Manager Cloud Control Administrator's Guide*.

Example 5-1 XML Format of Events

```
<EMEvent xmlns="http://xmlns.oracle.com/sysman/connector">
  <ExternalEventID/>
  <SystemAttributes>
    <EventClass/>
    <OccurredDate/>
    <ReportedDate/>
    <EventName/>
    <Severity/>
    <SeverityCode/>
    <SourceInfo>
      <TargetInfo>
        <TargetName/>
        <TargetTypeLabel/>
        <TargetProperty>
          <Name/>
          <Value/>
        </TargetProperty>
      </TargetInfo>
    </SourceInfo>
    <Message/>
    <EventURL/>
  </SystemAttributes>
  <EventContextAttributes>
    <NumberAttribute>
      <Name/>
      <Value/>
    </NumberAttribute>
    <StringAttribute>
      <Name/>
      <Value/>
    </StringAttribute>
  </EventContextAttributes>
</EMEvent>
```

Table 5-5 provides a description of the fields shown in Example 5-1.

Table 5-5 Field Descriptions for XML Format

Field	Description
ExternalEventID	Unique identifier of the alert in IBM Netcool/OMNIBus. This will only be set for updates.
EventClass	Identifies the type of event.
OccurredDate	In cases where the event information is mined from a log, this field contains the date and time when the event occurred based on the entry in the log.
ReportedDate	Date/time the event was reported by Enterprise Manager.
EventName	Descriptive name of the event.
Severity	Severity of the event: Fatal, Critical, Warning, Minor_Warning, Informational, or Clear.
SeverityCode	Severity code for the event: FATAL, CRITICAL, WARNING, MINOR_WARNING, INFORMATIONAL, or CLEAR.
TargetName	Target name that is a unique instance of the target type.
TargetTypeLabel	Display name of the target type for which the event was generated.
TargetProperty	Additional target properties that do not have a specific field in the event model (name:value pair).
Message	Description of the event.
EventURL	Link to the web page for the event.
EventContextAttributes	Additional event properties that do not have a specific field in the event model (name:value pair).

Managing Templates in Oracle Enterprise Manager 12c

Oracle Enterprise Manager 12c (Release 12.1.0.2 and later) includes the functionality to allow you to manage templates within Cloud Control. Templates, which are included as part of the connector installation, are automatically registered and are shown within Enterprise Manager.

If you need to use new a template that is not included as part of the connector installation, then you will need to register it using the `emctl` command. See [Manually Changing a Mapping](#) for more information. Once you have registered the template, you will be able to manage it through Enterprise Manager.

Using Enterprise Manager, you can edit a template, view the XML code, delete a template, and restore a deleted template.

To begin:

1. Log in to Enterprise Manager Cloud Control.
2. From the Setup menu, select **Extensibility**, then **Management Connectors**.
3. On the Management Connectors page, select the **IBM Tivoli Netcool/OMNIBus Connector**. Click **Configure**.
4. On the Configure Management Connector page, click the **Template** tab.

**Note:**

This feature is available on Oracle Enterprise Manager Release 12.1.0.2.0 or later only. To verify your installed release version:

1. Log in to Enterprise Manager.
2. From the Help menu, select **About Cloud Control**.
3. The pop-up window displays the installed version of Oracle Enterprise Manager.

Edit a Template with Enterprise Manager

To edit an existing template:

1. Log in to Enterprise Manager Cloud Control.
2. From the Setup menu, select **Extensibility**, then **Management Connectors**.
3. On the Management Connectors page, select the **IBM Tivoli Netcool/OMNibus Connector**. Click **Configure**.
4. On the Configure Management Connector page, click the **Template** tab.
5. Select a template and click **Edit**.
6. After you edit the Description or Template Contents fields, click **Save**. On the confirmation page, click **Yes** to complete the template save.

View Template Code in Enterprise Manager

To view template code:

1. Log in to Enterprise Manager Cloud Control.
2. From the Setup menu, select **Extensibility**, then **Management Connectors**.
3. On the Management Connectors page, select the **IBM Tivoli Netcool/OMNibus Connector**. Click **Configure**.
4. On the Configure Management Connector page, click the **Template** tab.
5. Select a template and click **View**.

The event templates are in XSLT format. A basic knowledge of XSLT is required to understand the code.

Delete a Template with Enterprise Manager

To delete a template:

1. Log in to Enterprise Manager Cloud Control.
2. From the Setup menu, select **Extensibility**, then **Management Connectors**.
3. On the Management Connectors page, select the **IBM Tivoli Netcool/OMNibus Connector**. Click **Configure**.
4. On the Configure Management Connector page, click the **Template** tab.

5. Select a template and click **Delete**.
6. Click **Yes** to confirm the delete operation.

Restore a Deleted Template with Enterprise Manager

To restore a deleted template:

1. Log in to Enterprise Manager Cloud Control.
2. From the Setup menu, select **Extensibility**, then **Management Connectors**.
3. On the Management Connectors page, select the **IBM Tivoli Netcool/OMNIBus Connector**. Click **Configure**.
4. On the Configure Management Connector page, click the **Template** tab.
5. Select a template and click **Restore**.
6. Click **Yes** in the confirmation page that appears in order to restore.



Note:

You can restore out-of-box templates that you have may edited or customized to their default values. You cannot restore new templates that you have created (using the `emctl register_template` command) described in [Manually Changing a Mapping](#).

Manually Changing a Mapping

The following procedure provides the steps required to change a mapping.

- Export the default template XSLT files from Enterprise Manager to use as a baseline. Follow the instructions in [Exporting the Template Files](#) to obtain copies of the default template files.
Create a custom template file by copying the XSL file you want to change to a new file with a unique name.
- Make the desired mapping changes to the template file. See [Modifying Templates](#) for guidance on mapping changes.
- Register the new template with Enterprise Manager as specified in [Registering Templates](#).
- See [Template Customization Example](#) for an example of how to customize templates.

Exporting the Template Files

The connector installation jar file is included in the Self Update archive that was installed in [Installing the Connector in Enterprise Manager](#).

To extract the installation file:

1. Copy the `omnibus_connector.jar` file from the Management Server host system to a temporary directory. The file is located in the archives directory that was extracted in [Exporting the Installation Files](#).

2. Extract the templates from the connector jar file using the following command:

```
$JAVA_HOME/bin/jar xf omnibus_connector.jar createEvent_request.xml  
updateEvent_request.xml
```

 **Note:**

If the system where the connector file is located does not have the JDK installed, you cannot extract the jar file contents. You must copy the jar file to a system that has the JDK installed and transfer the template files after they have been extracted.

Modifying Templates

A template is an XSLT file that transforms an input XML format to an output XML format. To make the mapping changes, you need to understand the input XML format being transformed and the output XML the template generates. The input document is the XML document that contains the Enterprise Manager event information. The output document is the request being sent to the web service for Netcool/OMNIbus. The output XML document must conform to the data format specified in the Netcool/OMNIbus EventService WSDL as shown in [Web Service for Netcool/OMNIbus EventService WSDL Format](#).

[Enterprise Manager Event Format](#) contains details on the format of the event data generated by Enterprise Manager. [Web Service for Netcool/OMNIbus EventService WSDL Format](#) contains details on the XML format required by the web service for Netcool/OMNIbus EventService. [Mappings Between XML Format and Message Field Names](#) in [Web Service for Netcool/OMNIbus Details](#) provides the mapping between the web service for Netcool/OMNIbus EventService fields and the field names in Netcool/OMNIbus. Listed below are the steps you should follow to modify the template file.

To modify a template:

1. Review [Mappings Between XML Format and Message Field Names](#) in [Web Service for Netcool/OMNIbus Details](#) and [Web Service for Netcool/OMNIbus EventService WSDL Format](#) to identify the names of the fields in the web service for Netcool/OMNIbus that you are planning to modify. For example, if you want to populate the Node field in Netcool/OMNIbus, look up the field in [Web Service for Netcool/OMNIbus Details](#) and observe that the web service for Netcool/OMNIbus expects the data in the `/create/event/source/displayName` XML path.
2. If you are planning on using data from the event, review [Enterprise Manager Event Format](#) to identify the field(s) you want to use in the mapping. [Enterprise Manager Event Examples](#) provides sample transactions that could be very helpful in identifying what data is present in the XML document from Enterprise Manager.
3. Open the custom template file in a text editor or a utility that can be used to build XSLT files.
4. Make the necessary changes to the custom template file and save it.
5. Unless the change is very basic, you should test the changes that you have made using the sample data from [Enterprise Manager Event Examples](#). This requires a utility that performs XSLT translations.

Registering Templates

You need to reregister your customized templates before they are recognized in Enterprise Manager.

For each customized template, run the following `emctl register_template connector` command. The command must be run as a user that has execute privileges on `emctl` and read privileges on the template:

```
$ORACLE_HOME/bin/emctl register_template connector
-t <template.xml> -repos_pwd <reposPassword>
-cname <connectorName> -iname <internalName>
-tname <templateName> -ttype 2 [-d <description>]
```

Where:

<template.xml> is the full path name of the custom template file.

<reposPassword> is the Enterprise Manager root password.

<connectorName> is the name of the connector instance installed in step 5 of [Creating a Connector Instance](#).

<internalName> is the internal name to use for the template.

<templateName> is the name to display for the template.

<description> is the description of the template.

For example, the following command updates the `createEvent` template for the connector instance with the name of "Netcool/OMNIBus:"

```
emctl register_template connector
-t /home/oracle/templates/createEvent_custom_request.xml
-repos_pwd sysmanpass
-cname "Netcool/OMNIBus"
-iname "createEvent"
-tname "Create Event Request"
-ttype 2
-d "Demo template"
```

Template Customization Example

In this scenario, the customer has determined that the default mapping does not work for their environment. Instead of placing target name information in the `Node` field, the customer would like to place target type information in the `Node` field. The customer would also like to set the `Location` field if the target properties have location information specified. They have created a temporary directory on the OMS server system at `/home/oracle/customized_templates` to contain the templates.

The following steps explain how the customer would customize the templates to meet their requirements.

1. Open a command window on the OMS server system and change the working directory to the temporary directory:

```
cd /home/oracle/customized_templates
```

2. Copy the `omnibus_connector.jar` file from the archives directory from [Exporting the Installation Files](#) to the temporary directory:

```
cp $exported_dir/archives/omnibus_connector.jar .
```

3. Extract the template files from the jar file. This command extracts the `createEvent` and `updateEvent` template files:

```
$JAVA_HOME/bin/jar xf omnibus_connector.jar createEvent_request.xml
updateEvent_request.xml
```

4. Make a working copy of the template files for modification:


```
cp createEvent_request.xml createEvent_request_custom.xml
```
5. Open the `createEvent_request_custom.xml` file in a text editor.
6. Change the mapping for the `Node` field as shown below.

Before Changes

The following code shows the `Node` mapping in the file before the changes:

```
<source>
  <name>
    <!-- Sets the Netcool/OMNIBus NodeAlias field -->
    <xsl:value-of select="a:SystemAttributes/a:SourceInfo/a:TargetInfo/
a:TargetName"/><xsl:value-of select="a:SystemAttributes/a:SourceInfo/a:TargetInfo/
a:TargetProperty[a:Name='Target_Host']/a:Value"/>
  </name>
  <displayName>
    <!-- Sets the Netcool/OMNIBus Node field -->
    <xsl:value-of select="a:SystemAttributes/a:SourceInfo/a:TargetInfo/
a:TargetName"/>
  </displayName>
</source>
```

After Changes

The following code shows the `Node` mapping after the changes. The changes are shown in bold:

```
<source>
  <name>
    <!-- Sets the Netcool/OMNIBus NodeAlias field -->
    <xsl:value-of select="a:SystemAttributes/a:SourceInfo/a:TargetInfo/
a:TargetName"/><xsl:value-of select="a:SystemAttributes/a:SourceInfo/a:TargetInfo/
a:TargetProperty[a:Name='Target_Host']/a:Value"/>
  </name>
  <displayName>
    <!-- Sets the Netcool/OMNIBus Node field -->
    <xsl:value-of select="a:SystemAttributes/a:SourceInfo/a:TargetInfo/
a:TargetTypeLabel" />
  </displayName>
</source>
```

7. Add logic to populate the `Location` field if the target properties contain location information.

Before Changes

The following code shows the extended fields mapping in the file before the changes:

```
<extended-fields>
  <!-- Sets the Netcool/OMNIBus Manager field -->
```

```

    <string-field name="Manager">Oracle Enterprise Manager</string-field>
    ...
</extended-fields>

```

After Changes

The following code shows the `extended` fields mapping after the changes. The changes are shown in bold:

```

<extended-fields>
  <xsl:variable name="location">
    <xsl:for-each select="a:SystemAttributes/a:SourceInfo/a:TargetInfo/
a:TargetProperty">
      <xsl:if test="a:Name = 'Location'">
        <xsl:value-of select="a:Value"/>
      </xsl:if>
    </xsl:for-each>
  </xsl:variable>
  <!-- Sets the Netcool/OMNIBus Location field if specified in the target
properties -->
  <xsl:if test="$location != ''">
    <string-field name="Location"><xsl:value-of select="$location"/> </
string-field>
  </xsl:if>
  <!-- Sets the Netcool/OMNIBus Manager field -->
  <string-field name="Manager">Oracle Enterprise Manager</string-field>
  ...
</extended-fields>

```

8. Save your changes after making the updates.
9. To set up for testing, create an empty `create.xml` file and open in a text editor.
10. Copy the contents of the XML document in the [Create Example](#) section of [Enterprise Manager Event Examples](#) into the `create.xml` file and save it.
11. Test the new templates using a utility that performs XSLT translations using the `create.xml` file as the input XML document.

Note:

There are a variety of off-the-shelf testing utilities that are available. Use the utility that works best for your environment.

12. Examine the output XML document and verify that the results are correct.

Note:

You can also use an XML client to send the XML output document to the web service for Netcool/OMNIBus to verify that it is a format accepted by the web service.

13. If the XML document has problems, make any necessary corrections and repeat steps 11 and 12.
14. At the OMS server system, run the following command to re-register your customized `createEvent` template:

```
emctl register_template connector
-t /home/oracle/customized_templates/createEvent_request_custom.xml
-repos_pwd sysmanpass
-cname "Netcool/OMNIBus"
-iname "createEvent"
-tname "Create Event Request"
-ttype 2
-d "This is the customized request xml file for the createEvent method"
```

6

Enabling SSL for HTTPS

This chapter provides the following procedures for configuring the IBM Tivoli Netcool/OMNIBus Connector to use the HTTPS protocol:

- [Configure Oracle Enterprise Manager to Use SSL](#)
- [Installing a Self-Signed Certificate](#)
- [Installing a Certificate from a Certified Authority](#)
- [Importing the Web Service Certificate into Enterprise Manager](#)

Configure Oracle Enterprise Manager to Use SSL

If the web service for Netcool/OMNIBus was configured to run using the HTTPS protocol, you must perform the following steps to set up SSL:

1. Install an SSL certificate in the web service for Netcool/OMNIBus keystore. You must either install a self-signed certificate or install a certificate obtained from a Certificate Authority (CA).
 - To install a self-signed certificate, perform the steps specified in [Installing a Self-Signed Certificate](#).
 - To install a certificate from a CA, perform the steps specified in [Installing a Certificate from a Certified Authority](#).
2. Restart the web service for Netcool/OMNIBus for the changes to take effect.
3. Import the SSL certificate from the web service for Netcool/OMNIBus keystore into the Enterprise Manager keystore as specified in [Importing the Web Service Certificate into Enterprise Manager](#).

Installing a Self-Signed Certificate

To generate and install a self-signed SSL certificate for the Netcool/OMNIBus adapter web service:

1. Open a command prompt window and change the working directory to the `adapters/conf` directory in the web service for Netcool/OMNIBus installation directory.
2. Enter the following command to delete the default SSL entry from the web service Netcool/OMNIBus keystore:

- **Unix**

```
$JAVA_HOME/bin/keytool -delete -alias iwave -keypass iwavepw -storepass iwavepw -keystore keystore.jks
```

- **Windows**

```
"%JAVA_HOME%\bin\keytool" -delete -alias iwave -keypass iwavepw -storepass iwavepw -keystore keystore.jks
```

3. Enter the following command to generate a new certificate and place it in the web service for Netcool/OMNIBus keystore. You will need to replace `<hostname>` with the host name or IP address of the system where the web service for Netcool/OMNIBus is installed.

 **Note:**

The host name in the certificate must match the host name or IP address the web service uses. If they do not match, a failure occurs when Enterprise Manager attempts to invoke the web service.

- **Unix**

```
$JAVA_HOME/bin/keytool -genkey -alias iwave -keyalg RSA -keysize 1024 -  
dname "CN=<hostname>, OU=Development, O=iWave Software, L=Frisco, ST=TX,  
C=US" -keypass iwavepw -storepass iwavepw -keystore keystore.jks
```

- **Windows**

```
"%JAVA_HOME%\bin\keytool" -genkey -alias iwave -keyalg RSA -keysize 1024  
-dname "CN=<hostname>, OU=Development, O=iWave Software, L=Frisco,  
ST=TX, C=US" -keypass iwavepw -storepass iwavepw -keystore keystore.jks
```

Installing a Certificate from a Certified Authority

To request and install a signed SSL certificate for the web service for Netcool/OMNIBus:

1. Request a certificate for the web service for Netcool/OMNIBus from a Certificate Authority, such as VeriSign.

 **Note:**

In the certificate request, make sure to specify the host name or IP address of the system where the web service for Netcool/OMNIBus is installed. The host name in the certificate must match the host name or IP address the web service uses. If they do not match, a failure occurs when Enterprise Manager attempts to invoke the web service.

2. After you obtain the certificate from the Certificate Authority, open a command prompt window and change the working directory to the `adapters/conf` directory in the web service for Netcool/OMNIBus installation directory.
3. Enter the following command to delete the default SSL entry from the web service for Netcool/OMNIBus keystore:

- **Unix**

```
$JAVA_HOME/bin/keytool -delete -alias iwave -keypass iwavepw -storepass  
iwavepw -keystore keystore.jks
```

- **Windows**

```
"%JAVA_HOME%\bin\keytool" -delete -alias iwave -keypass iwavepw -  
storepass iwavepw -keystore keystore.jks
```


4. Enter the following command to install the certificate, where `<certificateFile>` is the full path name of the file provided by the Certificate Authority:

- Unix

```
$JAVA_HOME/bin/keytool -importcert -alias iwave -file <certificateFile> -
keypass iwavepw -storepass iwavepw -keystore keystore.jks
```

- Windows

```
"%JAVA_HOME%\bin\keytool" -importcert -alias iwave -file <certificateFile> -
keypass iwavepw -storepass iwavepw -keystore keystore.jks
```

Importing the Web Service Certificate into Enterprise Manager

To import the web service for Netcool/OMNIBus SSL certificate into the Enterprise Manager keystore:

1. Open a command prompt window and change the working directory to the `adapters/conf` directory in the web service for Netcool/OMNIBus installation directory.
2. Issue the following command to extract the SSL certificate from the web service Netcool/OMNIBus keystore and place it in the `OMNIBusws.cer` certificate file:

- Unix

```
$JAVA_HOME/bin/keytool -exportcert -rfc -alias iwave -file OMNIBusws.cer -
keystore keystore.jks -storepass iwavepw
```

- Windows

```
"%JAVA_HOME%\bin\keytool" -exportcert -rfc -alias iwave -file OMNIBusws.cer -
keystore keystore.jks -storepass iwavepw
```

3. Transfer the certificate file `OMNIBusws.cer` to the system where Enterprise Manager is installed.
4. Append the contents of the `OMNIBusws.cer` file to:

```
$INSTANCE_HOME/sysman/config/b64LocalCertificate.txt
```

5. Ensure that only the following lines are appended to the `b64LocalCertificate.txt` file (that is, do not include blank lines, comments, or any other special characters):

```
-----BEGIN CERTIFICATE-----
<<<Certificate in Base64 format>>>
-----END CERTIFICATE-----
```

6. Restart OMS by running the following commands:

```
emctl stop oms
emctl start oms
```

Note:

Do not run the `emctl secure oms/agent` command after adding the external certificate to the `b64LocalCertificate.txt` file. If you run the `emctl secure` command later, then repeat steps 4 through 6 to make sure the external certificate exists in the `b64LocalCertificate.txt` file.

7

Troubleshooting

The Oracle Enterprise Manager Connector Framework requires a web service interface for exchanging event information with Netcool/OMNIbus. Since Netcool/OMNIbus does not come with a web services front end, an Oracle-provided web service front end must be installed before Oracle Enterprise Manager can exchange event information with Netcool/OMNIbus. Additionally, an Oracle Agent for Netcool/OMNIbus must also be installed.

This chapter provides information to assist in troubleshooting integration issues with IBM Netcool/OMNIbus. The chapter focuses on troubleshooting issues in the web service front-end and the Oracle Agent.

This chapter discusses the following topics:

- [Diagnosing the Problem](#)
- [Troubleshooting Web Service Startup Errors](#)
- [JVM Errors](#)
- [Using the Correct URL for Web Service Operations](#)
- [Troubleshooting Web Service Operations Errors](#)
- [Resolving Errors from Oracle Enterprise Manager](#)

Before you start the troubleshooting process, ensure that you have done the following:

- Installed the IBM Tivoli Netcool/OMNIbus Connector as specified in [Installing the Connector in Enterprise Manager](#).
- Installed and started the Oracle Agent for Netcool/OMNIbus as specified in [Installing the Oracle Agent for Netcool/OMNIbus](#).
- Installed, started, and tested the web service for Netcool/OMNIbus service as specified in [Installing the Web Service for Netcool/OMNIbus Front-End](#).
- Made the necessary modifications to Netcool/OMNIbus as specified in [Modifying Netcool/OMNIbus](#).
- Created a connector instance as specified in [Creating a Connector Instance](#).
- Configured the connector instance as specified in [Configuring the Connector](#).
- Set up one or more rules as specified in [Setting Event Rules](#) to forward events to the connector instance.

If all the actions above have been completed and the connector is not working, perform the steps in [Diagnosing the Problem](#).

Diagnosing the Problem

To diagnose the problem:

1. Verify that the web service for Netcool/OMNIbus has been successfully started and the WSDL for the web service can be accessed from the system where the web service for Netcool/OMNIbus is installed:

- a. Perform the steps in [Determining the URL Used by the Web Service](#) to determine the URL used by the web service for Netcool/OMNIBus.
- b. Open a browser and paste the URL from the previous step in the address window. Append `?WSDL` to the end of the URL and attempt to load the URL.
- c. The WSDL should be loaded if the Adapter is operational.

 **Note:**

A WSDL is an XML file that describes the web service.

- d. If the WSDL cannot be loaded, this indicates the web service had startup issues. See [Troubleshooting Web Service Startup Errors](#) to diagnose the problem.
2. Verify that the WSDL for the web service for Netcool/OMNIBus can be accessed from the system where the Enterprise Manager server is installed:
 - a. Open a browser at the Enterprise Manager server and copy the URL from step 1-b above to the address window. The host name for the URL will be `localhost`. Change `localhost` to the actual host name or IP address of the system where the web service for Netcool/OMNIBus is installed. If you specify a host name, make sure that the host name is recognized at the Enterprise Manager server system. This can be done using the `ping` command.

For example, if the web service for Netcool/OMNIBus is installed on the server with a host name of `SDServer01` and the URL listed in `framework.log` is:

```
https://localhost:8443/services/omnibus/EventService
```

The URL used at the Enterprise Manager server system would be:

```
https://OMServer01:8443/services/omnibus/EventService
```

- b. Attempt to load the WSDL by appending `?WSDL` at the end of the URL. If the WSDL cannot be loaded, either the host name is not recognized at the Enterprise Manager system, or there is a connectivity issue between the two systems. If you specified a host name, try using the IP address instead of the host name in the URL. If it still does not load, you have a connectivity problem. You will need to consult with your IT department to resolve this issue.
3. Verify that the Netcool/OMNIBus Connector specifies the correct URL for the `createEvent` and `updateEvent` operations:
 - a. Log in to Oracle Enterprise Manager Cloud Control with an account that has Super Administrator permissions.
 - b. From the **Setup** menu, select **Extensibility**, then **Management Connectors**.
The Management Connectors page appears, which shows the installed connectors.
 - c. Click the **Configure** icon associated with the Netcool/OMNIBus Connector. This invokes edit mode, enabling you to configure the connector.
 - d. Verify that the modified URL from step 2-a is specified for both operations (`createEvent` and `updateEvent`). If any of the operations are incorrect, change to the correct URL.

 **Note:**

Do not append the WSDL to the end of the URL that is specified here.

- e. Click **OK**.

If no errors were found in the previous steps, this rules out connectivity issues between the connector and the web service for Netcool/OMNIbus. The problem must originate in the web service or the Oracle Agent. See [Troubleshooting Web Service Operations Errors](#) for information on diagnosing errors.

Troubleshooting Web Service Startup Errors

To identify the cause of a startup failure, navigate to the `adapters/log` directory in the web service for Netcool/OMNIbus installation directory and open the `framework.log` file in a text editor. Search for **Exception** to find any errors in the file. If the file does not exist, it indicates that there is a problem locating or executing the JVM. See [JVM Errors](#) for information about resolving JVM issues.

Listed below are some possible Exceptions, an explanation of the root cause, and a description of the solution.

java.net.BindException: Address already in use: bind

This error indicates that the web service could not start because of a port conflict. There are two possible causes for this error:

- Another application is using a port that the web service is configured to use. If the web service is configured to use SSL, the port number is 8443. If it is not configured to use SSL, the port number is 8080.

There are two possible solutions to this. You can change the other application to use a different port or you can change the web service for Netcool/OMNIbus to use a different port. To change the web service to use a different port, see [Changing Default Port Numbers](#) in [Web Service for Netcool/OMNIbus Details](#).

- An instance of the web service is already running. If this is the case, no change is required. You should only run one instance of the web service at a time.

org.springframework.beans.factory.BeanInitializationException: Could not load properties; nested exception is java.io.FileNotFoundException: ... framework.properties (Permission denied)

This error indicates that the web service could not start because the permissions on the `framework.properties` file in the `conf` directory were not set correctly.

To solve the problem, change the permissions to give the account or group under which the web service for Netcool/OMNIbus runs read and execute permissions.

org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'com.iwave.mapping.MappingGroovyAdapter#1' defined in class path resource [com/iwave/adapters/omnibus/event-adapter-synchronous.xml]: Initialization of bean failed; nested exception is org.springframework.beans.TypeMismatchException: Failed to convert property value of type [java.lang.String] to required type [java.lang.Class]

for property 'toType'; nested exception is java.lang.IllegalArgumentException: Cannot find class [com.tivoli.tec.event_delivery.TECEvent]

This error indicates that the `evd.jar` file is missing.

To solve the problem, perform the steps in [Copying the Required .jar Files](#) and then attempt to start the web service for Netcool/OMNIbus.

org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'omnibusAdapterConfig' defined in class path resource [com/iwave/adapters/omnibus/event-adapter-synchronous.xml]: Invocation of init method failed; nested exception is com.iwave.adapters.omnibus.InvalidAdapterConfigurationException: Error loading the JDBC driver (com.sybase.jdbc2.jdbc.SybDriver) specified in the properties file: com.sybase.jdbc2.jdbc.SybDriver

This error indicates that the `jconn2.jar` file is missing.

To solve the problem, perform the steps in [Copying the Required .jar Files](#) and then attempt to start the web service.

**Note:**

For any other startup errors, consult with Oracle Support.

JVM Errors

The web service for Netcool/OMNIbus requires version 1.6 of the JVM. If multiple versions of the JVM are installed on the system, it is possible that an older version of the JVM is being executed whenever the web service starts.

On Unix systems, the `JAVA_HOME` environment variable must be set to the directory where Java Development Kit (JDK) 1.6 is installed in the shell where the web service is started. To properly start the web service on a Unix platform:

1. Set the `JAVA_HOME` environment variable to the JDK 1.6 installation directory.
2. Navigate to the `adapters/bin` sub-directory in the web service installation directory.
3. Execute the `./service.sh start` command.

On Windows systems, perform the following to ensure that JDK 1.6 is used when starting the web service:

1. Navigate to the `adapters/bin` sub-directory in the web service installation directory.
2. Run the `iWaveAdaptersw.exe` executable.
3. Click the **Java** tab.
4. Make sure that the Use Default check box is not checked.
5. In the Java Virtual system box, specify the path to the `jvm.dll` file in the JDK 1.6 installation directory.

6. Click **OK**.

Using the Correct URL for Web Service Operations

Perform the following steps to identify and configure the connector to use the correct URL for web service for Netcool/OMNIBus operations:

1. Perform the steps in section [Determining the URL Used by the Web Service](#) to obtain the URL used by the web service.
2. Replace localhost in the URL with the hostname or IP address of the system where the web service is installed.
3. Log in to Oracle Enterprise Manager Cloud Control with an account that has Super Administrator permissions.
4. From the **Setup** menu, select **Extensibility**, then **Management Connectors**.
The General tab of the Configure Management Connector page appears.
5. Click on the name of the Netcool/OMNIBus connector.
6. Verify that the URL identified in step 2 is specified for the `createEvent` and `updateEvent` operations.
7. If any of the operations are incorrect, change to the correct URL and click **OK**.

Determining the URL Used by the Web Service

Perform the following steps to identify the correct URL for web service for Netcool/OMNIBus operations:

1. Open a command terminal on the system where the web service is installed.
2. Change the working directory to the `adapters/log` directory in the web service installation directory.
3. Open the `framework.log` file in a text editor.
4. Go to the bottom of the file and search backwards for the string `Setting the server's publish address to be`. Continue searching backwards until you find the URL that contains `EventService`. The URL listed here is the URL that the web service uses to handle create and update operations.

Troubleshooting Web Service Operations Errors

Perform the following diagnostic steps if messages are not being generated or updated as expected in Netcool/OMNIBus:

1. Verify that the event that was triggered is referenced in a rule that forwards events to the IBM Tivoli Netcool/OMNIBus Connector.
2. Determine the error that Oracle Enterprise Manager has reported by examining the log file for errors. Perform the following steps to locate errors in the log file:
 - a. Open the `emoms_pbs.trc` file in a text editor. The file is located in the `log` directory at the following location:

```
<EM_INSTANCE_BASE>/em/<OMS_NAME>/sysman/log/
```

... where `<EM_INSTANCE_BASE>` is the OMS Instance Base directory. By default, the OMS Instance Base directory is `gc_inst`, which is present under the parent directory of the Oracle Middleware Home.

For example, if the Oracle Middleware Home is `/u01/app/Oracle/Middleware`, then the instance base directory is `/u01/app/Oracle/Middleware/gc_inst`, and the log and trace files are available in the `/u01/app/Oracle/Middleware/gc_inst/em/EMGC_OMS1/sysman/log/` directory path.

- b. Go to the bottom of the file and search backwards for `Caused by`.

Lines that start with `Caused by` contain error information. The error information appears after the text in the line that reads:

```
oracle.sysman.emSDK.webservices.outbound.WSInvokeException: caught
WebServiceException:
```

3. Diagnose the problem based on the error information. See [Resolving Errors from Oracle Enterprise Manager](#) for information on troubleshooting common error events.

Resolving Errors from Oracle Enterprise Manager

This section provides cause and solution information on troubleshooting common error messages. Find the error message in [Table 7-1](#) that matches your error message, then refer to the corresponding section(s) indicated under Possible Cause for instructions on diagnosing and correcting the problem.

Table 7-1 Oracle Enterprise Manager Error Messages

Error Message	Possible Cause
<pre>javax.xml.soap.SOAPException: javax.xml.soap.SOAPException: Bad response: 403 Forbidden from url</pre>	Invalid Web Service Credentials
<pre>jjavax.xml.soap.SOAPException: javax.xml.soap.SOAPException: Message send failed: sun.security.validator.ValidatorException: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuil derException: unable to find valid certification path to requested target</pre>	SSL Not Configured in Enterprise Manager
<pre>javax.xml.soap.SOAPException: javax.xml.soap.SOAPException: Message send failed: Connection refused</pre>	Web Service is Down
<pre>javax.xml.soap.SOAPException: javax.xml.soap.SOAPException: Message send failed: No route to host</pre>	Invalid IP Address

Table 7-1 (Cont.) Oracle Enterprise Manager Error Messages

Error Message	Possible Cause
<pre>javax.xml.soap.SOAPException: javax.xml.soap.SOAPException: Bad response: 404 Not Found from url ...</pre>	Invalid URL Path
<pre>javax.xml.soap.SOAPException: javax.xml.soap.SOAPException: Message send failed: Connection timed out</pre>	Firewall Blocking Access or Invalid Port Number
<pre>javax.xml.soap.SOAPException: javax.xml.soap.SOAPException: Message send failed: <hostname></pre>	Unknown Host
<pre>javax.xml.transform.TransformerConfigurationException: Could not compile stylesheet</pre>	Invalid XML Format
<pre>com/ibm/logging/ILogger</pre>	Missing Log Jar
<pre>Timeout occurred waiting for synchronous response from OMNIBus after inserting an event.</pre>	Timeout
<pre>Attempt to insert the event into the master probe failed.</pre>	Cannot Connect to the Probe

Invalid Web Service Credentials

Cause

The user name or password for accessing the web service for Netcool/OMNIBus is incorrect.

Solution

1. Log in to Oracle Enterprise Manager Cloud Control with an account that has Super Administrator privileges.
2. From the **Setup** menu, select **Extensibility**, then **Management Connectors**.
The Management Connectors page appears.
3. Click the name of the Netcool/OMNIBus Connector.
This action invokes Edit mode, enabling you to configure the connector.
4. Correct the **Netcool/OMNIBus Web Service Username** and **Netcool/OMNIBus Web Service Password** fields and click **OK**.
5. If the credentials supplied appear to have been entered correctly but still do not work, a typographical error might have been made during the setup of the web service. The

recommended option is to reset the web service credentials and attempt this again. Refer to [Changing Web Service Credentials](#) in [Web Service for Netcool/OMNIbus Details](#) for the steps required to reset the credentials.

SSL Not Configured in Enterprise Manager

Cause

The SSL handshake between the Oracle Enterprise Manager Connector Framework and the web service for Netcool/OMNIbus has failed. This failure occurs because Oracle Enterprise Manager is not configured correctly with the SSL certificate for the web service. The SSL certificate that the web service uses must be imported into the Enterprise Manager key store. The certificate is either missing from the key store or does not match the SSL certificate provided by the web service for Netcool/OMNIbus.

Solution

Import the SSL certificate from the web service into the Enterprise Manager key store. See [Configure Oracle Enterprise Manager to Use SSL](#) for details on setting up Oracle Enterprise Manager with the Netcool/OMNIbus SSL certificate.

Web Service is Down

Cause

The web service for Netcool/OMNIbus is down.

Solution

Perform the following steps to check the status of the web service and start it if necessary.

If the web service is installed on a Unix system:

1. Open a command terminal on the system where the web service is installed.
2. Change the working directory to the `adapters/bin` directory in the web service installation directory.
3. Enter the following command:

```
./service.sh status
```
4. If the command indicates that the service is not running, restart the web service as specified in [Running the Web Service on Unix](#).

If the web service is installed on a Windows system:

1. Open a command terminal on the system where the web service is installed.
2. Change the working directory to the `adapters\log` directory in the web service installation directory.
3. Open the `framework.log` file in a text editor.
4. Go to the bottom of the file and search backwards for the string `iWave Adapter Framework`. If the last occurrence found is `iWave Adapter Framework Started`, this indicates that the web service is started.

5. If the web service is not started, restart the web service as specified in [Running the Web Service on Windows](#).

Invalid IP Address

Cause

The IP address specified in the URL is invalid, or the network is down.

Solution

Verify that the hostname/IP address configured for the connector is correct:

1. Log in to Oracle Enterprise Manager Cloud Control with an account that has Super Administrator privileges.
2. From the **Setup** menu, select **Extensibility**, then **Management Connectors**.
The Management Connectors page appears, which shows the installed connectors.
3. Click on the name of the Netcool/OMNIBus Connector. This invokes edit mode, enabling you to configure the connector.
4. Verify that the hostname/IP address specified in the URL for the `createEvent` and `updateEvent` operations are correct.
5. If the hostname/IP address is incorrect, provide the correct values and click **OK**.

If the URLs specify a host name, make sure that the host name resolves to the correct IP address. To determine the IP address of the host name, issue the `ping <hostname>` command, where `<hostname>` is the actual host name. This lists the IP address that was resolved for the host name. If this is incorrect, the system administrator needs to investigate why it is incorrect.

If the hostname/IP address appears to be correct, try to ping the system where the web service is installed using the hostname/IP address. If the ping fails, the system administrator needs to investigate why there is no connectivity.

Invalid URL Path

Cause

The web service received the request and rejected it because an invalid path was specified in the URL.

Solution

Perform the following steps to test the URL the connector is using:

1. Log in to Oracle Enterprise Manager Cloud Control with an account that has Super Administrator privileges.
2. From the **Setup** menu, select **Extensibility**, then **Management Connectors**.
The Management Connectors page appears, which shows the installed connectors.
3. Click on the name of the Netcool/OMNIBus Connector. This invokes edit mode, enabling you to configure the connector.
4. Select and copy the URL specified for the `createEvent` operation.

5. Open an internet browser on the system where the Oracle Enterprise Manager server is installed.
6. In the address window, enter the URL that was copied in step 6 above. Add `?wsdl` to the end of the URL. The URL should appear similar to the following example:

```
http://[Hostname]:8080/services/hpovou/EventService?wsdl
```

[Hostname] is the actual host name or IP address where the web service is installed.

If the WSDL is loaded, this confirms that the URL is correct. If it fails to load, there is a problem with the URL. Perform the steps specified in [Using the Correct URL for Web Service Operations](#) to configure the connector to use the correct URL.

Firewall Blocking Access

Cause

A firewall is blocking access to the system where the web service for Netcool/OMNIbus is installed.

Solution

Contact your IT department to give Enterprise Manager access to the port the web service uses. Perform the steps specified in [Using the Correct URL for Web Service Operations](#) to determine the URL used by the Web Service. The port number specified in the URL is the port number the IT department should open in the firewall.

Invalid Port Number

Cause

The port number specified in the URL is invalid.

Solution

Verify that the hostname/IP address configured for the connector is correct:

1. Log in to Oracle Enterprise Manager Cloud Control with an account that has Super Administrator privileges.
2. From the **Setup** menu, select **Extensibility**, then **Management Connectors**.
The Management Connectors page appears, which shows the installed connectors.
3. Click on the name of the Netcool/OMNIbus Connector. This invokes edit mode, enabling you to configure the connector.
4. Verify that the port number specified in the URL for the `createEvent` and `updateEvent` operations are correct.

If the web service is configured to use SSL, the port number should be **8443**. Otherwise, the port number should be **8080** unless the port number has been reconfigured as specified in [Changing Default Port Numbers in Web Service for Netcool/OMNIbus Details](#).

5. If the port number is incorrect, provide the correct value and click **OK**.

Unknown Host

Cause

The system does not recognize the host name specified in the URL.

Solution

You have the following options for addressing this issue:

- Coordinate with the system administrator to change the system configuration to recognize the host name.
- Change the host name if it is listed incorrectly.
- Specify the IP address in the URL instead of the host name. To do this, perform the following steps:
 1. Determine the IP address of the system where the web service is installed.
 2. Log in to Oracle Enterprise Manager Cloud Control with an account that has Super Administrator privileges.
 3. From the **Setup** menu, select **Extensibility**, then **Management Connectors**.
The Management Connectors page appears, which shows the installed connectors.
 4. Click on the name of the Netcool/OMNIBus Connector. This invokes edit mode, enabling you to configure the connector.
 5. Change the host name to the IP address in the URL specified for the `createEvent` and `updateEvent` operations.
 6. Click **OK**.

Make sure that the host name resolves to the correct IP address. To determine the IP address of the host name, issue the `ping <hostname>` command, where `<hostname>` is the actual host name. This lists the IP address that was resolved for the host name. If this is incorrect, the system administrator needs to investigate why it is incorrect.

If the hostname/IP address appears to be correct, try to ping the system where the web service is installed using the hostname/IP address. If the ping fails, the system administrator needs to investigate why there is no connectivity.

Invalid XML Format

Cause

The connector framework could not process the request because the XSL file was formatted incorrectly. This problem should not occur unless the connector has been customized.

Solution

Examine any changes made to the XSL template files for mistakes that could have caused the problem. If you cannot find the problem manually, load the XSL in a utility that performs XML validation.

Missing Log Jar

Cause

The error indicates that the web service cannot locate the `log.jar` file.

Solution

To solve the problem, perform the steps in [Copying the Required .jar Files](#) and then restart the web service for Netcool/OMNIBus.

Timeout

Cause

The web service received the request and successfully sent the request to the TEC EIF probe. The web service timed out waiting for Netcool/OMNIBus to send a notification that the requested alert that was created/updated. This error occurs because Netcool/OMNIBus is not properly configured to send the response to the web service.

Listed below is a summary of the sequence of events that occur whenever an event is created/updated in Netcool/OMNIBus.

1. Enterprise Manager makes a web service call for a create/update operation.
2. The web service makes an API call to the EIF probe to insert an event.
3. The EIF Probe inserts a row in the `custom.oracle_status` table for the event.
4. The `oracle_insert_alert` trigger fires and creates/updates the appropriate alert in the `alerts.status`, `alerts.details`, and `alerts.journal` database tables.
5. The `oracle_alert_created`, `oracle_alert_reinserted`, or `oracle_alert_updated` trigger fires calling the `oracle_send` external procedure.
6. The `oracle_send` external procedure calls the `oracle_send.cmd` (Windows) or `oracle_send.sh` (Unix) script to send a notification to the web service.
7. The `oracle_send.cmd/oracle_send.sh` script calls the universal agent utility (`uniagt.exe`) to send the notification to the web service.
8. Upon receipt of the notification, the web service makes a JDBC call to retrieve field information for the alert.
9. The web service sends a response to Enterprise Manager with the alert information.

A failure in steps 3 through 8 results in a timeout being reported by the web service for Netcool/OMNIBus.

Solution

Perform the following steps to identify the location of the failure:

1. Verify that Netcool/OMNIBus has been customized as specified in [Modifying Netcool/OMNIBus](#) and that the Object Server and EIF Probe are operational. Correct any issues that are found and retry.

2. Open the Netcool/OMNIBus console and determine whether an alert was created/updated.

If the alert was NOT created/updated, the failure occurred in Cause step 3 or 4. Perform the following steps to determine the reason for the failure:

- a. Check the EIF Probe log file for errors. If there are errors, you will need to determine the cause of the error and fix the problem. Most likely the error will be caused by a lack of connectivity between the probe system and the object server system.
 - b. If there are no errors in the EIF Probe log file, verify that a row was added to the `custom.oracle_status` table for the event. If a row was not added, there must be a problem with the configuration of the EIF probe. Enable debug in the EIF Probe and run another test. Look in the EIF Probe log file to see if there is any information about what the probe did with the event.
 - c. If a row was added to the `custom.oracle_status` table, examine the `ErrorMessage` column for any error information.
 - d. Examine the object server log file to verify that the `oracle_insert_alert` trigger fired. If the trigger did not fire, open the definition for the trigger and verify it is enabled.
 - e. If the trigger fired, look for any errors that occurred while running the trigger.
3. If the alert was created/updated, the failure occurred somewhere in Cause steps 5 through 8. Perform the following steps to determine the reason for the failure:

- a. Examine the object server log file to verify that one of the `oracle_alert_created`, `oracle_alert_reinserted`, or `oracle_alert_updated` triggers fired. If none of the triggers fired, open the definition for the triggers and look for anything that might prevent the trigger from firing. One possibility is that the trigger has been disabled.
- b. If one of the triggers fired, look for any errors that occurred while running the trigger. If there are no errors in the object server log, you will need to enable debug in the script to see if it is being called. Step c below gives instructions for enabling debug in a Windows environment and step d gives instructions for enabling debug in a Unix environment.
- c. For Windows environments, perform the following steps to enable debug in the notification script.

Open the `oracle_send.cmd` script with a text editor.

Uncomment the following lines by removing the **REM** at the beginning of the line:

```
echo %DATE% %TIME% %* >> "%AGENT_DIR%\oracle_send.out"
"%AGENT_DIR%\bin\uniagt" -r %OMNIBUS_WS_URL% -v 1 -O "%AGENT_DIR%\agent%1.log"
```

Comment the following line by prepending **REM** to the beginning of the line:

```
"%AGENT_DIR%\bin\uniagt" -r %OMNIBUS_WS_URL%
```

Save the file and exit.

- d. For Unix environments, perform the following steps to enable debug in the notification script:

Open the `oracle_send.sh` script with a text editor.

Uncomment the following lines by removing the **#** at the beginning of the line:

```
echo `date` $* >> ${AGENT_DIR}/oracle_send.out  
  
${AGENT_DIR}/bin/uniagt -r ${OMNIBUS_WS_URL} -v 1 -O ${AGENT_DIR}/agent/  
{1}.log
```

Comment the following line by prepending # to the beginning of the line:

```
${AGENT_DIR}/bin/uniagt -r ${OMNIBUS_WS_URL}
```

Save the file and exit.

- e. After enabling debug, run another test and determine whether the `oracle_send.out` log file was created. If the file was created, open it with a text editor and look at the last line in the file to get the serial number. The line starts with the date/time the script was called followed by the serial number and the name of the trigger that fired.
- f. Verify that an agent log was created for the serial number. The agent log should be named `agent<serialno>.log` where `<serialno>` is the serial number from the `oracle_send.out` log file.
- g. If the file exists, open it with a text editor and look for errors sending to the web service. If errors occur, it will likely be caused by a connectivity issue or an invalid URL specified in the `OMNIBUS_WS_URL` environment variable. Check the hostname/IP address and port number in the URL. Unless specifically changed, the port number should be **8080**. If the notification was sent successfully, you will see something similar to the following in the log.

```
2012/06/13 11:20:52 response headers: <HTTP/1.1 200 OK  
Content-Type: text/xml  
Content-Length: 0  
Server: Jetty(6.1.14)
```

- h. If the agent log indicates the notification was successfully sent to the web service, look in the web service log for errors. The web service log is located in the `adapters/log` directory of the web service installation directory and is named `framework.log`. Most likely the error will be caused by a failure to connect to the database to retrieve the alert information.

Cannot Connect to the Probe

Cause

The web service for Netcool/OMNibus cannot connect to the TEC EIF probe and send alert information. Listed below are some of the possible issues that could cause a connection failure.

- The probe is down.
- The hostname/IP address or the port number for the probe is configured incorrectly in the web service.
- A firewall is preventing the connection from completing.

Solution

Perform the following steps to resolve the issue:

1. Verify that the TEC EIF probe is up and operational.

2. Verify that the web service is configured to use the correct hostname/IP address and port number.
To verify:
 - a. Open the `framework.properties` file using a text editor. The file is located in the `adapters/conf` directory of the web service installation directory.
 - b. Verify that the `omnibus.probe.master.t_ServerLocation` property is set to the hostname or IP address where the master EIF probe is installed.
 - c. Verify that the `omnibus.probe.master.t_Port` property is set to the port number on which the master probe is listening.
 - d. If configured to send to a slave probe, verify that the `omnibus.probe.slave.t_ServerLocation` property is set to the hostname or IP address where the slave EIF probe is installed.
 - e. If configured to send to a slave probe, verify that the `omnibus.probe.slave.t_Port` property is set to the port number on which the slave probe is listening.
 - f. Make any required changes and save the file.
 - g. If any changes were made, restart the web service as specified in the appropriate section ([Running the Web Service on Unix](#) or [Running the Web Service on Windows](#)).
3. If the problem still is not resolved, contact your IT department to check for a firewall that might be preventing the web service from calling the EIF probe.

A

Web Service for Netcool/OMNIbus Details

This appendix describes the Netcool/OMNIbus mappings and various other configuration defaults in the following sections:

- [XML Format of Netcool/OMNIbus Alerts](#)
- [Changing Default Port Numbers](#)
- [Changing Web Service Credentials](#)

XML Format of Netcool/OMNIbus Alerts

[Example A-1](#) represents the format that the web service for Netcool/OMNIbus expects for creating new alerts in Netcool/OMNIbus. The format for update requests is the same, except that the root node would be update instead of create.

Example A-1 Sample Create Format for the Web Service for Netcool/OMNIbus

```
<iwaveaf:create xmlns:iwaveaf="http://iwavesoftware.com/services/adaptor-framework">
  <event>
    <identifier></identifier>
    <source>
      <name></name>
      <displayName></displayName>
    </source>
    <metric>
      <category></category>
    </metric>
    <object>
      <identifier></identifier>
    </object>
    <severity></severity>
    <description></description>
    <type></type>
    <logs>
      <log>
        <extended-fields>
          <string-field name="1"></string-field>
          <string-field name="2"></string-field>
          <string-field name="3"></string-field>
          <string-field name="4"></string-field>
          <string-field name="5"></string-field>
          <string-field name="6"></string-field>
          <string-field name="7"></string-field>
          <string-field name="8"></string-field>
          <string-field name="9"></string-field>
          <string-field name="10"></string-field>
          <string-field name="11"></string-field>
          <string-field name="12"></string-field>
          <string-field name="13"></string-field>
          <string-field name="14"></string-field>
          <string-field name="15"></string-field>
          <string-field name="16"></string-field>
        </extended-fields>
      </log>
    </logs>
  </event>
</iwaveaf:create>
```

```

        </extended-fields>
    </log>
</logs>
<extended-fields>
  <string-field name="Manager"> </string-field>
  <string-field name="Agent"></string-field>
  <date-field name="LastOccurrence"></date-field>
  <string-field name="URL"></string-field>
  <string-field name="Detail_1_Name"> </string-field>
  <string-field name="Detail_1_Value"> </string-field>
  <string-field name="Detail_2_Name"></string-field>
  <string-field name="Detail_2_Value"></string-field>
  <string-field name="Detail_3_Name"></string-field>
  <string-field name="Detail_3_Value"> </string-field>
  <string-field name="Detail_4_Name"></string-field>
  <string-field name="Detail_4_Value"> </string-field>
  <string-field name="Detail_5_Name"></string-field>
  <string-field name="Detail_5_Value"></string-field>
  <string-field name="Detail_6_Name"></string-field>
  <string-field name="Detail_6_Value"></string-field>
  <string-field name="Detail_7_Name"></string-field>
  <string-field name="Detail_7_Value"> </string-field>
  <string-field name="Detail_8_Name"></string-field>
  <string-field name="Detail_8_Value"></string-field>
  <string-field name="Detail_9_Name"></string-field>
  <string-field name="Detail_9_Value"></string-field>
  <string-field name="Detail_10_Name"></string-field>
  <string-field name="Detail_10_Value"></string-field>
  <string-field name="Detail_11_Name"></string-field>
  <string-field name="Detail_11_Value"></string-field>
  <string-field name="Detail_12_Name"></string-field>
  <string-field name="Detail_12_Value"></string-field>
  <string-field name="Detail_13_Name"></string-field>
  <string-field name="Detail_13_Value"></string-field>
  <string-field name="Detail_14_Name"></string-field>
  <string-field name="Detail_14_Value"></string-field>
  <string-field name="Detail_15_Name"></string-field>
  <string-field name="Detail_15_Value"></string-field>
  <string-field name="Detail_16_Name"></string-field>
  <string-field name="Detail_16_Value"></string-field>
  <string-field name="Detail_17_Name"></string-field>
  <string-field name="Detail_17_Value"></string-field>
  <string-field name="Detail_18_Name"></string-field>
  <string-field name="Detail_18_Value"></string-field>
  <string-field name="Detail_19_Name"></string-field>
  <string-field name="Detail_19_Value"></string-field>
  <string-field name="Detail_20_Name"></string-field>
  <string-field name="Detail_20_Value"></string-field>
</extended-fields>
</event>
</iwaveaf:create>

```

Mappings Between XML Format and Message Field Names

[Table A-1](#) identifies the mappings between the IBM Tivoli Netcool/OMNIBus alert field names (from the `alerts.status` table) and the XML format that the web service uses. To set the IBM Tivoli Netcool/OMNIBus alert fields, the XML document presented to the web service must have the corresponding fields set. This must be handled in the appropriate translation file identified in [Table 5-1](#).

 **Note:**

The XML paths specified in this table are for create requests. To determine the XML path for update requests, replace the **create** node in the path with **update**.

For example, the XML path for the `Identifier` field on an update operation would be:

```
/update/event/identifier
```

Table A-1 Event Fields and XML Path Mappings

Netcool/OMNIBus Event Field	Web Service XML Path
Identifier	/create/event/identifier
NodeAlias	/create/event/source/name
Node	/create/event/source/displayName
AlertGroup	/create/event/metric/category
AlertKey	/create/event/object/identifier
Severity	/create/event/severity
Summary	/create/event/description
Type	/create/event/type
Journal	See Log Entries below
Manager	See Extended Fields below
Agent	See Extended Fields below
LastOccurrence	See Extended Fields below
URL	See Extended Fields below

Log Entries

The `alerts.journal` table is populated from data passed in the `<logs>` element. An entry is added to the journal table for each `<log>` element that is a child of the `<logs>` element. The text for the journal entry is passed in `<string-field/>` elements that are children of `<extended-fields>` elements within the `<log>` element.

The name attribute in the `<string-field/>` element must be set to an integer from **1** to **16**. A journal entry has 16 columns that can contain text, and the name attribute determines what column in the journal table to populate. The column is 255 characters in length; so, strings longer than 255 characters will be truncated.

Extended Fields

An extended field is defined as a `<string-field/>` or `<date-field/>` element that is a child of the `<extended-fields>` node. Fields that contain date information are passed as a `date-field` element and all other fields are passed as a `string-field` element. The name of the extended field is specified in the name attribute, and the value of this field is specified as the

element value. The name must be the name of a database column in the `alerts.status` table or a special reserved name used to populate the `alert.details` table.

The web service allows up to 20 entries to be populated in the `alerts.details` table. To populate an entry, you must specify 2 elements. The `Detail_n_Name` element populates the `Name` field of the `alerts.details` table and the `Detail_n_Value` element populates the `Detail` field. For both elements, the `n` would be replaced by an integer between **1** and **20**. The number identifies the position where the information would appear on the Alert Details tab of the Event Information window.

Listed below is an example that shows how to populate fields in the `alerts.status`, `alerts.details`, and `alerts.journal` database tables. In the example, the `Manager` and `LastOccurrence` fields would be populated in the `alerts.status` table for the row with an identifier of `TESTYA3F17890220E040020A78012524`. The lines that start with **Detail** would cause the first two rows of the `alerts.details` table to be populated for the alert with an identifier of `TESTYA3F17890220E040020A78012524`. The log entry will cause a journal entry to be added to the `alerts.journal` table for the alert with an ID of `TESTYA3F17890220E040020A78012524`.

Example A-2 How to Populate Fields

```
<event>
  <identifier>TESTYA3F17890220E040020A78012524</identifier>
  <logs>
    <log>
      <extended-fields>
        <string-field name="1">This creates an entry in the journal</
xsl:text></string-field>
      </extended-fields>
    </log>
  </logs>
  <extended-fields>
    <string-field name="Manager">Oracle Enterprise Manager</string-field>
    <date-field name="LastOccurrence">2012-01-13T16:08:45.000-06:00</date-
field>
    <string-field name="Detail_1_Name">Reported Date:</string-field>
    <date-field name="Detail_1_Value">2012-05-03T00:00:00.0-00:00</date-field>
    <string-field name="Detail_2_Name"> Event Class:</string-field>
    <string-field name="Detail_2_Value">metric_alert</string-field>
  </extended-fields>
</event>
```

Changing Default Port Numbers

In most cases, you can use the default port numbers that the web service uses. However, if there are any conflicts with existing applications, you need to change the port numbers.

The following sections provide procedures on how to change these default port numbers:

- [Changing the Web Service Port \(8080\)](#)
- [Changing the Web Service Port \(8443\)](#)

Changing the Web Service Port (8080)

The web service for Netcool/OMNIBus uses port 8080 as the default port for communication when the web service is configured at installation to use HTTP (no SSL). To change the port number to a different value:

1. Identify a new port number to use for the web service. You must verify that the new port number is not in use on the system where the web service is installed.
2. Change the web service to use the new port number:
 - a. At the system where the web service is installed, navigate to the `adapters/conf` directory and open the `framework.properties` file in a text editor.
 - b. Change the port number from 8080 to the new port number for the following properties:

```
services.url
client.url
notification.url
acquisitionevent.url
```
 - c. Save and close the `framework.properties` file.
 - d. Stop and then start the web service as instructed in [Running the Web Service on Unix](#) and [Running the Web Service on Windows](#).
3. Change the configuration of the IBM Tivoli Netcool/OMNIBus Connector to use the new port number:
 - a. From the **Setup** menu of Oracle Enterprise Manager Cloud Control, select **Extensibility**, then **Management Connectors**.
 - b. Click on the name of the Netcool/OMNIBus Connector.
The General tab of the Configure Management Connector page appears.
 - c. Change the URLs listed in the **Web Service End Points** section to use the new port number.
 - d. Click **OK** to save your changes.
4. Change the agent scripts to send alert information to the new port number:
 - a. Navigate to the directory where the Oracle Agent for Netcool/OMNIBus is installed.
 - b. Open the `oracle_error.cmd` (Windows) or `oracle_error.sh` (Unix) script file in a text editor.
 - c. Change the port number in the `OMNIBUS_ADAPTER_URL` variable from **8080** to the new port number and save.
 - d. Open the `oracle_send.cmd` (Windows) or `oracle_send.sh` (Unix) script file in a text editor.
 - e. Change the port number in the `OMNIBUS_ADAPTER_URL` variable from **8080** to the new port number and save.

Changing the Web Service Port (8443)

Contact Oracle Support for assistance in switching the default SSL port 8443 to a different port.

Changing Web Service Credentials

Occasionally, problems occur when accessing the web service for Netcool/OMNIbus, because the credentials provided are incorrect. Whenever this happens, you receive an HTTP 403 error from the web service. You first want to check the specified credentials and verify that you entered the correct information. If the credentials you have specified appear to be correct, but you still receive the 403 error, the best option is to reset the web service credentials.

To reset the web service credentials:

1. Open a command prompt window and change the working directory to the web service `adapters/conf` directory using the following command:

```
cd <OMNIWS_INSTALL>/adapters/conf
```

Replace `<OMNIWS_INSTALL>` with the directory where the web service is installed.

2. Enter the following command to change the user name and password for accessing the web service:

```
../bin/propertiesEditor.sh -e  
framework.username="<username>" -h  
framework.password="<password>" framework.properties
```

... where `<username>` is the user name to specify for the web service, and `<password>` is the password.

The `propertiesEditor.sh` script is specifically for the Unix platform. The equivalent script for Windows platforms is `propertiesEditor.bat`.

3. Stop and then start the web service as instructed in [Running the Web Service on Unix](#) and [Running the Web Service on Windows](#).

B

Enterprise Manager Event Format

Templates translate the XML format of Enterprise Manager events to the XML format required for the endpoint application's web service. To customize your templates, you need to understand the format of both ends so that you know how to map the data. This appendix provides detailed information on the format of the XML document that contains event information from Enterprise Manager.

A schema file defines the structure of an XML document. The EMEvent.xsd schema file defines the main structure of the event data. This file also imports structure information from other schema files.

The following sections provide information from the schema files that define the EMEvent XML format:

- [EMEvent.xsd](#)
- [connectorCommon.xsd](#)
- [externalEvent.xsd](#)

EMEvent.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns="http://xmlns.oracle.com/sysman/connector"
            targetNamespace="http://xmlns.oracle.com/sysman/connector"
            elementFormDefault="qualified"
            xmlns:jaxb="http://java.sun.com/xml/ns/jaxb"
            jaxb:version="2.0"
            xmlns:xjc="http://java.sun.com/xml/ns/jaxb/xjc"
            jaxb:extensionBindingPrefixes="xjc">

  <xsd:annotation>
    <xsd:appinfo>
      <jaxb:globalBindings>
        <xjc:simple />
      </jaxb:globalBindings>
    </xsd:appinfo>
  </xsd:annotation>
  <xsd:include schemaLocation="connectorCommon.xsd"/>
  <xsd:element name="EMEvent" type="EMEventType" />
  <xsd:complexType name="EMEventType">
    <xsd:sequence>
      <xsd:element name="ConnectorGUID" type="xsd:string"/>
      <xsd:element name="ExternalEventID" type="xsd:string"
        minOccurs="0"/>
      <xsd:element name="NotificationRuleOwner" type="xsd:string"/>
      <xsd:element name="NotificationRuleName" type="xsd:string"/>
      <xsd:element name="ConnectorVariable" type="VariableType"
        minOccurs="0" maxOccurs="50"/>
      <xsd:element name="Property" type="PropertyType" minOccurs="0"
        maxOccurs="50"/>
      <xsd:element name="SystemAttributes"
```

```
        type="EventSystemAttributesType"/>
<xsd:element name="EventClassSpecificAttributes">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:choice minOccurs="0" maxOccurs="200">
        <xsd:element name="StringAttribute" type="StringValueType"/>
        <xsd:element name="NumberAttribute" type="StringValueType"/>
        <xsd:element name="RawAttribute" type="StringValueType"/>
        <xsd:element name="DateAttribute" type="DateValueType"/>
      </xsd:choice>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="EventContextAttributes">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:choice minOccurs="0" maxOccurs="200">
        <xsd:element name="StringAttribute" type="StringValueType"/>
        <xsd:element name="NumberAttribute" type="StringValueType"/>
      </xsd:choice>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="EventSystemAttributesType">
  <xsd:sequence>
    <xsd:element name="EventClass" type="xsd:string"/>
    <xsd:element name="EventID" type="xsd:string"/>
    <xsd:element name="SequenceID" type="xsd:string"/>
    <xsd:element name="OccurredDate" type="xsd:dateTime" minOccurs="0"/>
    <xsd:element name="ReportedDate" type="xsd:dateTime"/>
    <xsd:element name="DisplayTZ" type="xsd:string"/>
    <xsd:element name="EventName" type="xsd:string"/>
    <xsd:element name="Severity" type="xsd:string"/>
    <xsd:element name="SeverityCode" type="xsd:string"/>
    <xsd:element name="SourceInfo" type="SourceInfoType"/>
    <xsd:element name="Message" type="xsd:string" minOccurs="0"/>
  </xsd:sequence>
<xsd:element name="ActionMessage" type="xsd:string" minOccurs="0"/>
<xsd:element name="EventURL" type="xsd:string"/>
<xsd:element name="AutoClose" type="xsd:boolean"/>
<xsd:element name="EventCategory" type="xsd:string" minOccurs="0"
  maxOccurs="50"/>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="StringValueType">
  <xsd:sequence>
    <xsd:element name="Name" type="xsd:string"/>
    <xsd:element name="Value" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="DateValueType">
  <xsd:sequence>
    <xsd:element name="Name" type="xsd:string"/>
    <xsd:element name="Value" type="xsd:dateTime"/>
  </xsd:sequence>
</xsd:complexType>
</xsd:schema>
```


connectorCommon.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://xmlns.oracle.com/sysman/connector"
  targetNamespace="http://xmlns.oracle.com/sysman/connector"
  elementFormDefault="qualified">
  <xsd:include schemaLocation="externalEvent.xsd"/>
  <xsd:complexType name="SourceInfoType">
    <xsd:annotation>
      <xsd:documentation>
        This section defines a complex type for Source Information.
      </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="SourceObjInfo" type="SourceObjInfoType"
minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>
            This element defines the data structure for the source object, the
            EM subsystem or component, that raises an EM event or an incident.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="TargetInfo" type="TargetInfoType"
minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>
            The element defines the data structure for an EM target as related
            to the connector framework.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="SourceObjInfoType">
    <xsd:annotation>
      <xsd:documentation>
        This section defines a complex type for Source Object Information.
      </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="ObjID" type="xsd:string">
        <xsd:annotation>
          <xsd:documentation>
            The unique ID to identify the source object.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="ObjName" type="xsd:string">
        <xsd:annotation>
          <xsd:documentation>
            The name of the source object.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="ObjOwner" type="xsd:string" minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>

```



```
</xsd:annotation>
</xsd:element>
<xsd:element name="TargetProperty" type="PropertyType" minOccurs="0"
  maxOccurs="50">
  <xsd:annotation>
    <xsd:documentation>
      An optional list of properties for the target.
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="PropertyType">
  <xsd:annotation>
    <xsd:documentation>
      This section defines a complex type for a property attribute.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="Name" type="xsd:string">
      <xsd:annotation>
        <xsd:documentation>
          A string name defining a property attribute.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="Value" type="xsd:string" nillable="true">
      <xsd:annotation>
        <xsd:documentation>
          A non-null string value.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="VariableType">
  <xsd:annotation>
    <xsd:documentation>
      This section defines a complex type for a general variable.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="VariableName" type="StringStrictT32">
      <xsd:annotation>
        <xsd:documentation>
          Name of the variable. It has to be a string containing 1 or upto
          32 upper case or lower case letters or numbers.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="VariableValue" type="StringT2048">
      <xsd:annotation>
        <xsd:documentation>
          Value of the variable. It has to be a string containing 1 or upto
          2048 characters.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="GetAlertsResponse">
```

```

    <xsd:annotation>
      <xsd:documentation>
        This section defines a complex type for responses to a getAlerts
request.
      </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="Alert" minOccurs="0"
maxOccurs="200">
        <xsd:annotation>
          <xsd:documentation>
            The individual alerts contained in the response. A response may
            have up to 200 alerts.
          </xsd:documentation>
        </xsd:annotation>
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element ref="ExternalEvent">
              <xsd:annotation>
                <xsd:documentation>
                  Details of the external event in the alert, as defined in
                  ExternalEvent.xsd.
                </xsd:documentation>
              </xsd:annotation>
            </xsd:element>
            <xsd:element name="InstanceVariable" type="VariableType"
              minOccurs="0" maxOccurs="50">
              <xsd:annotation>
                <xsd:documentation>
                  A list of instance variables for the alert.
                </xsd:documentation>
              </xsd:annotation>
            </xsd:element>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="ConnectorVariablesType">
    <xsd:annotation>
      <xsd:documentation>
        This section defines a complex type for connector variables. An
element
of type ConnectorVariablesType may have up to 50 connector variables,
as
defined next.
      </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="ConnectorVariable" type="VariableType" minOccurs="0"
        maxOccurs="50">
        <xsd:annotation>
          <xsd:documentation>
            A connector variabls as a name/value pair.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:simpleType name="StringT64">
    <xsd:annotation>

```

```
<xsd:documentation>
  This section defines a simple type for a String with maximum length of
  64 bytes.
</xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
  <xsd:minLength value="1"/>
  <xsd:maxLength value="64"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="StringT128">
  <xsd:annotation>
    <xsd:documentation>
      This section defines a simple type for a String with maximum length of
      128 bytes.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string">
    <xsd:minLength value="1"/>
    <xsd:maxLength value="128"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="StringT256">
  <xsd:annotation>
    <xsd:documentation>
      This section defines a simple type for a String with maximum length of
      256 bytes.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string">
    <xsd:minLength value="1"/>
    <xsd:maxLength value="256"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="StringT512">
  <xsd:annotation>
    <xsd:documentation>
      This section defines a simple type for a String with maximum length of
      512 bytes.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string">
    <xsd:minLength value="1"/>
    <xsd:maxLength value="512"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="StringT2048">
  <xsd:annotation>
    <xsd:documentation>
      This section defines a simple type for a String with maximum length of
      2048 bytes.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string">
    <xsd:minLength value="1"/>
    <xsd:maxLength value="2048"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="StringStrictT16">
  <xsd:annotation>
    <xsd:documentation>
```

```

        This section defines a simple type for a String with maximum length of
        16 bytes. The String can only contain lower or upper case letters,
        numbers, and the underscore characters.
    </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
    <xsd:minLength value="1"/>
    <xsd:maxLength value="16"/>
    <xsd:pattern value="([a-zA-Z0-9_])*"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="StringStrictT32">
    <xsd:annotation>
        <xsd:documentation>
            This section defines a simple type for a String with maximum length of
            32 bytes. The String can only contain lower or upper case letters,
numbers,
            and the underscore characters.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
        <xsd:minLength value="1"/>
        <xsd:maxLength value="32"/>
        <xsd:pattern value="([a-zA-Z0-9_])*"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="StringStrictT64">
    <xsd:annotation>
        <xsd:documentation>
            This section defines a simple type for a String with maximum length of
            64 bytes. The String can only contain lower or upper case letters,
            numbers, and the underscore characters.]
        </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
        <xsd:minLength value="1"/>
        <xsd:maxLength value="64"/>
        <xsd:pattern value="([a-zA-Z0-9_])*"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="StringStrictT128">
    <xsd:annotation>
        <xsd:documentation>
            This section defines a simple type for a String with maximum length of
            128 bytes. The String can only contain lower or upper case letters,
            numbers, and the underscore characters.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
        <xsd:minLength value="1"/>
        <xsd:maxLength value="128"/>
        <xsd:pattern value="([a-zA-Z0-9_])*"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="StringStrictT256">
    <xsd:annotation>
        <xsd:documentation>
            This section defines a simple type for a String with maximum length of
            256 bytes. The String can only contain lower or upper case letters,
            numbers, and the underscore characters.
        </xsd:documentation>

```

```

</xsd:annotation>
<xsd:restriction base="xsd:string">
  <xsd:minLength value="1"/>
  <xsd:maxLength value="256"/>
  <xsd:pattern value="([a-zA-Z0-9_]*)"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="VersionT">
  <xsd:annotation>
    <xsd:documentation>
      This section defines a simple type for a String with maximum length of
      20 bytes. The String can only contain numbers and the period characters.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string">
    <xsd:minLength value="1"/>
    <xsd:maxLength value="20"/>
    <xsd:pattern value="([0-9.]*)"/>
  </xsd:restriction>
</xsd:simpleType>
</xsd:schema>

```

externalEvent.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://xmlns.oracle.com/sysman/connector"
  targetNamespace="http://xmlns.oracle.com/sysman/connector"
  elementFormDefault="qualified">
  <xsd:element name="ExternalEvent">
    <xsd:annotation>
      <xsd:documentation>
        This section defines the attribute requirement of an external event
        for the connector framework to process it.
      </xsd:documentation>
    </xsd:annotation>
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="SystemAttributes"
          type="ExternalEventSystemAttributesType">
          <xsd:annotation>
            <xsd:documentation>
              Attributes to capture general information about the external event
              system. These attributes are system-specific, with all events from
              the same external system sharing the same system attributes.
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="EventClassAttributes"
          type="ExternalEventClassAttributesType">
          <xsd:annotation>
            <xsd:documentation>
              Attributes to capture specific information required for the event
              as defined in the event class.
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

```

```
<xsd:complexType name="ExternalEventSystemAttributesType">
  <xsd:annotation>
    <xsd:documentation>
      This section defines a complex type for system attributes required
for
      all external events.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="eventName" type="xsd:string"
minOccurs="0">
      <xsd:annotation>
        <xsd:documentation>
          Name of the event.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="category" type="xsd:string" minOccurs="0"
maxOccurs="50">
      <xsd:annotation>
        <xsd:documentation>
          The event category to which the event belongs.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="targetName" type="xsd:string">
      <xsd:annotation>
        <xsd:documentation>
          Name of the target on which event was generated. It refers
          to an entity in external systems similar to an EM target.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="targetType" type="xsd:string">
      <xsd:annotation>
        <xsd:documentation>
          The type of the target. Target types defined for event connectors
          are used. See connectorDeploy.xsd.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="occurrenceDate" type="xsd:dateTime">
      <xsd:annotation>
        <xsd:documentation>
          Date when the event occurred.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="detectedDate" type="xsd:dateTime">
      <xsd:annotation>
        <xsd:documentation>
          Date when the event was last detected.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="autoClose" type="xsd:boolean">
      <xsd:annotation>
        <xsd:documentation>
          A flag indicating if an event is auto closed by the system, or it
          has to be manually closed by users.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
```



```
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="message" type="xsd:string">
        <xsd:annotation>
            <xsd:documentation>
                A description of the event.
            </xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="severity" type="xsd:string">
        <xsd:annotation>
            <xsd:documentation>
                Severity level of the event.
            </xsd:documentation>
        </xsd:annotation>
    </xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ExternalEventClassAttributesType">
    <xsd:annotation>
        <xsd:documentation>
            This section defines a complex type for class specific attributes
            required for all external events in the class.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
<xsd:element name="external_event_id" type="xsd:string">
    <xsd:annotation>
        <xsd:documentation>
            ID used in external system to identify the event.
        </xsd:documentation>
    </xsd:annotation>
</xsd:element>
    <xsd:element name="external_rule_id" type="xsd:string"
minOccurs="0">
    <xsd:annotation>
        <xsd:documentation>
            Optional rule ID that delivered the event in the external system.
        </xsd:documentation>
    </xsd:annotation>
</xsd:element>
    <xsd:element name="external_host" type="xsd:string"
minOccurs="0">
    <xsd:annotation>
        <xsd:documentation>
            Optional host information from external system where event was
            generated.
        </xsd:documentation>
    </xsd:annotation>
</xsd:element>
    <xsd:element name="external_source" type="xsd:string"
minOccurs="0">
    <xsd:annotation>
        <xsd:documentation>
            Optional source information from the external system.
        </xsd:documentation>
    </xsd:annotation>
</xsd:element>
    <xsd:element name="external_severity" type="xsd:string">
    <xsd:annotation>
        <xsd:documentation>
```

```
        Severity level of the event on external system.
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="external_status" type="xsd:string"
minOccurs="0">
  <xsd:annotation>
    <xsd:documentation>
      Status of the event on extenral system.
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="custom_field1" type="xsd:string"
minOccurs="0">
  <xsd:annotation>
    <xsd:documentation>
      An optional field.
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="custom_field2" type="xsd:string"
minOccurs="0">
  <xsd:annotation>
    <xsd:documentation>
      An optional field.
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="custom_field3" type="xsd:string"
minOccurs="0">
  <xsd:annotation>
    <xsd:documentation>
      An optional field.
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="custom_field4" type="xsd:string"
minOccurs="0">
  <xsd:annotation>
    <xsd:documentation>
      An optional field.
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="custom_field5" type="xsd:string"
minOccurs="0">
  <xsd:annotation>
    <xsd:documentation>
      An optional field.
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:schema>
```

C

Web Service for Netcool/OMNIBus EventService WSDL Format

Templates translate the XML format of Enterprise Manager events to the XML format required for the endpoint application's web service. To customize your templates, you need to understand the format of both ends so that you know how to map the data. This appendix provides detailed information on the format of the XML document that the web service for Netcool/OMNIBus requires.

A WSDL file defines the interface that must be used to call a web service. The WSDL defines the structure of the XML document that it accepts and how the web service is to be called. The contents of the WSDL file are listed in [Example C-1](#).

Example C-1 EventService WSDL

```
<wsdl:definitions name="EventService"
  targetNamespace="http://iwavesoftware.com/services/adapter-framework"
  xmlns:ns1="http://schemas.xmlsoap.org/soap/http"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://iwavesoftware.com/services/adapter-framework"
  xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <wsdl:types>
    <xs:schema attributeFormDefault="unqualified"
      elementFormDefault="unqualified"
      targetNamespace="http://iwavesoftware.com/services/adapter-framework"
      xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:element name="Class" type="tns:eventClass"/>
      <xs:element name="ITObject" type="tns:itObject"/>
      <xs:element name="Object" type="tns:eventObject"/>
      <xs:element name="acknowledge" type="tns:acknowledge"/>
      <xs:element name="acknowledgeResponse" type="tns:acknowledgeResponse"/>
      <xs:element name="assignment" type="tns:assignment"/>
      <xs:element name="categorization" type="tns:categorization"/>
      <xs:element name="change" type="tns:change"/>
      <xs:element name="ci" type="tns:ci"/>
      <xs:element name="ciRef" type="tns:ciRef"/>
      <xs:element name="ciRelationship" type="tns:ciRelationship"/>
      <xs:element name="classification" type="tns:classification"/>
      <xs:element name="clear" type="tns:clear"/>
      <xs:element name="clearResponse" type="tns:clearResponse"/>
      <xs:element name="client-object" type="tns:clientDefinedObject"/>
      <xs:element name="complex-field" type="tns:complexField"/>
      <xs:element name="compliance" type="tns:compliance"/>
      <xs:element name="contact" type="tns:contact"/>
      <xs:element name="container" type="tns:container"/>
      <xs:element name="create" type="tns:create"/>
      <xs:element name="createResponse" type="tns:createResponse"/>
      <xs:element name="data" type="tns:data"/>
      <xs:element name="date-field" type="tns:dateField"/>
      <xs:element name="diary-list-field" type="tns:diaryListField"/>
      <xs:element name="diary-value" type="tns:diaryValue"/>
      <xs:element name="escalation" type="tns:escalation"/>
    </xs:schema>
  </wsdl:types>
</wsdl:definitions>
```

```

<xs:element name="event" type="tns:event"/>
<xs:element name="find" type="tns:find"/>
<xs:element name="findResponse" type="tns:findResponse"/>
<xs:element name="get" type="tns:get"/>
<xs:element name="getResponse" type="tns:getResponse"/>
<xs:element name="group" type="tns:group"/>
<xs:element name="incident" type="tns:incident"/>
<xs:element name="knowledge" type="tns:knowledge"/>
<xs:element name="log" type="tns:log"/>
<xs:element name="metric" type="tns:metric"/>
<xs:element name="number-field" type="tns:numberField"/>
<xs:element name="objectReference" type="tns:objectReference"/>
<xs:element name="problem" type="tns:problem"/>
<xs:element name="product" type="tns:product"/>
<xs:element name="resolution" type="tns:resolution"/>
<xs:element name="source" type="tns:source"/>
<xs:element name="string-field" type="tns:stringField"/>
<xs:element name="update" type="tns:update"/>
<xs:element name="updateResponse" type="tns:updateResponse"/>
<xs:element name="vendor" type="tns:vendor"/>
<xs:complexType name="clear">
  <xs:sequence>
    <xs:element minOccurs="0" name="event" type="tns:event"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="event">
  <xs:complexContent>
    <xs:extension base="tns:itObject">
      <xs:sequence>
        <xs:element default="1970-01-01T00:00:00.0-00:00" minOccurs="0"
name="createDate" type="xs:dateTime"/>
        <xs:element minOccurs="0" name="description" type="xs:string"/>
        <xs:element minOccurs="0" name="escalation" type="tns:escalation"/>
        <xs:element minOccurs="0" name="eventClass" type="tns:eventClass"/>
        <xs:element minOccurs="0" name="externalEventIdentifier"
type="xs:string"/>
        <xs:element minOccurs="0" name="externalIncidentIdentifier"
type="xs:string"/>
        <xs:element minOccurs="0" name="group" type="tns:group"/>
        <xs:element minOccurs="0" name="impact" type="xs:string"/>
        <xs:element minOccurs="0" name="knowledgeRoot">
          <xs:complexType>
            <xs:sequence>
              <xs:element maxOccurs="unbounded" minOccurs="0"
name="knowledge" type="tns:knowledge"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element default="1970-01-01T00:00:00.0-00:00" minOccurs="0"
name="lastModified" type="xs:dateTime"/>
        <xs:element minOccurs="0" name="lastModifiedBy" type="xs:string"/>
        <xs:element default="1970-01-01T00:00:00.0-00:00" minOccurs="0"
name="lastModifiedNonConnector" type="xs:dateTime"/>
        <xs:element default="1970-01-01T00:00:00.0-00:00" minOccurs="0"
name="lastRepeatDate" type="xs:dateTime"/>
        <xs:element minOccurs="0" name="logs">
          <xs:complexType>
            <xs:sequence>
              <xs:element maxOccurs="unbounded" minOccurs="0" name="log"
type="tns:log"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

        </xs:complexType>
    </xs:element>
    <xs:element default="1970-01-01T00:00:00.0-00:00" minOccurs="0"
name="maintenanceModeLastModified" type="xs:dateTime"/>
    <xs:element minOccurs="0" name="metric" type="tns:metric"/>
    <xs:element name="monitorEvent" type="xs:boolean"/>
    <xs:element minOccurs="0" name="object" type="tns:eventObject"/>
    <xs:element minOccurs="0" name="owner" type="xs:string"/>
    <xs:element minOccurs="0" name="priority" type="xs:string"/>
    <xs:element default="-2147483648" minOccurs="0" name="repeatCount"
type="xs:int"/>
    <xs:element minOccurs="0" name="resolvedBy" type="xs:string"/>
    <xs:element default="1970-01-01T00:00:00.0-00:00" minOccurs="0"
name="resolvedDate" type="xs:dateTime"/>
    <xs:element minOccurs="0" name="severity" type="xs:string"/>
    <xs:element minOccurs="0" name="source" type="tns:source"/>
    <xs:element minOccurs="0" name="status" type="xs:string"/>
    <xs:element default="1970-01-01T00:00:00.0-00:00" minOccurs="0"
name="statusLastModified" type="xs:dateTime"/>
    <xs:element minOccurs="0" name="summary" type="xs:string"/>
    <xs:element minOccurs="0" name="type" type="xs:string"/>
    <xs:element minOccurs="0" name="urgency" type="xs:string"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="itObject">
    <xs:sequence>
        <xs:element minOccurs="0" name="extended-fields">
            <xs:complexType>
                <xs:sequence>
                    <xs:choice maxOccurs="unbounded" minOccurs="0">
                        <xs:element name="string-field" type="tns:stringField"/>
                        <xs:element name="date-field" type="tns:dateField"/>
                        <xs:element name="number-field" type="tns:numberField"/>
                        <xs:element name="diary-list-field" type="tns:diaryListField"/>
                        <xs:element name="complex-field" type="tns:complexTypeField"/>
                    </xs:choice>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
        <xs:element minOccurs="0" name="identifier" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="escalation">
    <xs:sequence>
        <xs:element name="escalated" type="xs:boolean"/>
        <xs:element minOccurs="0" name="escalatedBy" type="xs:string"/>
        <xs:element minOccurs="0" name="server" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="eventClass">
    <xs:sequence>
        <xs:element minOccurs="0" name="container" type="tns:container"/>
        <xs:element minOccurs="0" name="description" type="xs:string"/>
        <xs:element minOccurs="0" name="displayName" type="xs:string"/>
        <xs:element minOccurs="0" name="identifier" type="xs:string"/>
        <xs:element minOccurs="0" name="name" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="container">

```

```

    <xs:sequence>
      <xs:element minOccurs="0" name="description" type="xs:string"/>
      <xs:element minOccurs="0" name="displayName" type="xs:string"/>
      <xs:element minOccurs="0" name="identifier" type="xs:string"/>
      <xs:element minOccurs="0" name="name" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="group">
    <xs:sequence>
      <xs:element minOccurs="0" name="identifier" type="xs:string"/>
      <xs:element minOccurs="0" name="name" type="xs:string"/>
      <xs:element minOccurs="0" name="version" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="knowledge">
    <xs:sequence>
      <xs:element minOccurs="0" name="content" type="xs:string"/>
      <xs:element default="1970-01-01T00:00:00.0-00:00" minOccurs="0"
name="dateCreated" type="xs:dateTime"/>
      <xs:element minOccurs="0" name="description" type="xs:string"/>
      <xs:element minOccurs="0" name="displayName" type="xs:string"/>
      <xs:element minOccurs="0" name="identifier" type="xs:string"/>
      <xs:element minOccurs="0" name="languageCode" type="xs:string"/>
      <xs:element default="1970-01-01T00:00:00.0-00:00" minOccurs="0"
name="lastModified" type="xs:dateTime"/>
      <xs:element minOccurs="0" name="name" type="xs:string"/>
      <xs:element minOccurs="0" name="status" type="xs:string"/>
      <xs:element minOccurs="0" name="summary" type="xs:string"/>
      <xs:element minOccurs="0" name="type" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="log">
    <xs:sequence>
      <xs:element minOccurs="0" name="extended-fields">
        <xs:complexType>
          <xs:sequence>
            <xs:choice maxOccurs="unbounded" minOccurs="0">
              <xs:element name="string-field" type="tns:stringField"/>
              <xs:element name="date-field" type="tns:dateField"/>
              <xs:element name="number-field" type="tns:numberField"/>
              <xs:element name="diary-list-field" type="tns:diaryListField"/>
              <xs:element name="complex-field" type="tns:complexType"/>
            </xs:choice>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element minOccurs="0" name="description" type="xs:string"/>
      <xs:element minOccurs="0" name="identifier" type="xs:string"/>
      <xs:element minOccurs="0" name="source" type="xs:string"/>
      <xs:element default="1970-01-01T00:00:00.0-00:00" minOccurs="0"
name="submitDate" type="xs:dateTime"/>
      <xs:element minOccurs="0" name="submitter" type="xs:string"/>
      <xs:element minOccurs="0" name="summary" type="xs:string"/>
      <xs:element minOccurs="0" name="type" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="stringField">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="name" type="xs:string" use="required"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

```

```

    </xs:simpleContent>
  </xs:complexType>
  <xs:complexType name="dateField">
    <xs:simpleContent>
      <xs:extension base="xs:dateTime">
        <xs:attribute name="name" type="xs:string" use="required"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:complexType name="numberField">
    <xs:simpleContent>
      <xs:extension base="xs:decimal">
        <xs:attribute name="name" type="xs:string" use="required"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:complexType name="diaryListField">
    <xs:sequence>
      <xs:element maxOccurs="unbounded" minOccurs="0" name="diary-value"
type="tns:diaryValue"/>
    </xs:sequence>
    <xs:attribute name="name" type="xs:string" use="required"/>
  </xs:complexType>
  <xs:complexType name="diaryValue">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="author" type="xs:string"/>
        <xs:attribute name="timestamp" type="xs:dateTime"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:complexType name="complexField">
    <xs:sequence>
      <xs:choice maxOccurs="unbounded" minOccurs="0">
        <xs:element name="string-field" type="tns:stringField"/>
        <xs:element name="date-field" type="tns:dateField"/>
        <xs:element name="number-field" type="tns:numberField"/>
        <xs:element name="diary-list-field" type="tns:diaryListField"/>
        <xs:element name="complex-field" type="tns:complexField"/>
      </xs:choice>
    </xs:sequence>
    <xs:attribute name="name" type="xs:string" use="required"/>
  </xs:complexType>
  <xs:complexType name="metric">
    <xs:sequence>
      <xs:element minOccurs="0" name="category" type="xs:string"/>
      <xs:element minOccurs="0" name="keyValues">
        <xs:complexType>
          <xs:sequence>
            <xs:element maxOccurs="unbounded" minOccurs="0" name="string"
type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element minOccurs="0" name="subCategory" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="eventObject">
    <xs:sequence>
      <xs:element minOccurs="0" name="computerName" type="xs:string"/>
      <xs:element minOccurs="0" name="displayName" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>

```

```

        <xs:element minOccurs="0" name="identifier" type="xs:string"/>
        <xs:element minOccurs="0" name="inMaintenanceMode" type="xs:boolean"/>
        <xs:element minOccurs="0" name="name" type="xs:string"/>
        <xs:element minOccurs="0" name="state" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="source">
    <xs:sequence>
        <xs:element minOccurs="0" name="computerName" type="xs:string"/>
        <xs:element minOccurs="0" name="container" type="tns:container"/>
        <xs:element minOccurs="0" name="description" type="xs:string"/>
        <xs:element minOccurs="0" name="displayName" type="xs:string"/>
        <xs:element minOccurs="0" name="domain" type="xs:string"/>
        <xs:element minOccurs="0" name="identifier" type="xs:string"/>
        <xs:element minOccurs="0" name="name" type="xs:string"/>
        <xs:element minOccurs="0" name="type" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="clearResponse">
    <xs:sequence/>
</xs:complexType>
<xs:complexType name="update">
    <xs:sequence>
        <xs:element minOccurs="0" name="event" type="tns:event"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="updateResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="tns:event"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="acknowledge">
    <xs:sequence>
        <xs:element minOccurs="0" name="event" type="tns:event"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="acknowledgeResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="tns:event"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="get">
    <xs:sequence>
        <xs:element minOccurs="0" name="identifier" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="tns:event"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="find">
    <xs:sequence>
        <xs:element minOccurs="0" name="query" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="findResponse">
    <xs:sequence>
        <xs:element maxOccurs="unbounded" minOccurs="0" name="return"
type="tns:event"/>
    </xs:sequence>

```



```

</xs:complexType>
<xs:complexType name="create">
  <xs:sequence>
    <xs:element minOccurs="0" name="event" type="tns:event"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="createResponse">
  <xs:sequence>
    <xs:element minOccurs="0" name="return" type="tns:event"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType mixed="true" name="clientDefinedObject">
  <xs:sequence>
    <xs:any maxOccurs="unbounded" minOccurs="0" namespace="##other"
processContents="skip"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="incident">
  <xs:complexContent>
    <xs:extension base="tns:itObject">
      <xs:sequence>
        <xs:element minOccurs="0" name="assignee" type="tns:assignment"/>
        <xs:element minOccurs="0" name="cis">
          <xs:complexType>
            <xs:sequence>
              <xs:element maxOccurs="unbounded" minOccurs="0" name="ci"
type="tns:ciRef"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element minOccurs="0" name="classification"
type="tns:classification"/>
        <xs:element minOccurs="0" name="customer" type="tns:contact"/>
        <xs:element minOccurs="0" name="description" type="xs:string"/>
        <xs:element minOccurs="0" name="direct" type="tns:contact"/>
        <xs:element minOccurs="0" name="impact" type="xs:string"/>
        <xs:element minOccurs="0" name="log-data">
          <xs:complexType>
            <xs:sequence>
              <xs:element maxOccurs="unbounded" minOccurs="0" name="log"
type="tns:log"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element minOccurs="0" name="logs">
          <xs:complexType>
            <xs:sequence>
              <xs:element maxOccurs="unbounded" minOccurs="0" name="log"
type="tns:objectReference"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element minOccurs="0" name="owner" type="tns:assignment"/>
        <xs:element minOccurs="0" name="priority" type="xs:string"/>
        <xs:element minOccurs="0" name="relationships">
          <xs:complexType>
            <xs:sequence>
              <xs:element maxOccurs="unbounded" minOccurs="0"
name="relationship" type="tns:objectReference"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

        </xs:element>
        <xs:element minOccurs="0" name="resolution" type="tns:resolution"/>
        <xs:element minOccurs="0" name="severity" type="xs:string"/>
        <xs:element minOccurs="0" name="status" type="xs:string"/>
        <xs:element minOccurs="0" name="summary" type="xs:string"/>
        <xs:element minOccurs="0" name="urgency" type="xs:string"/>
        <xs:element minOccurs="0" name="vendor" type="tns:vendor"/>
        <xs:element minOccurs="0" name="weight" type="xs:string"/>
    </xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="assignment">
    <xs:sequence>
        <xs:element minOccurs="0" name="company" type="xs:string"/>
        <xs:element minOccurs="0" name="organization" type="xs:string"/>
        <xs:element minOccurs="0" name="owner" type="xs:string"/>
        <xs:element minOccurs="0" name="primaryGroup" type="xs:string"/>
        <xs:element minOccurs="0" name="secondaryGroup" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ciRef">
    <xs:sequence>
        <xs:element minOccurs="0" name="extended-fields">
            <xs:complexType>
                <xs:sequence>
                    <xs:choice maxOccurs="unbounded" minOccurs="0">
                        <xs:element name="string-field" type="tns:stringField"/>
                        <xs:element name="date-field" type="tns:dateField"/>
                        <xs:element name="number-field" type="tns:numberField"/>
                        <xs:element name="diary-list-field" type="tns:diaryListField"/>
                        <xs:element name="complex-field" type="tns:complexTypeField"/>
                    </xs:choice>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
        <xs:element minOccurs="0" name="description" type="xs:string"/>
        <xs:element minOccurs="0" name="identifier" type="xs:string"/>
        <xs:element minOccurs="0" name="name" type="xs:string"/>
        <xs:element minOccurs="0" name="status" type="xs:string"/>
        <xs:element minOccurs="0" name="type" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="classification">
    <xs:sequence>
        <xs:element minOccurs="0" name="company" type="xs:string"/>
        <xs:element minOccurs="0" name="operational"
type="tns:categorization"/>
        <xs:element minOccurs="0" name="product" type="tns:product"/>
        <xs:element default="1970-01-01T00:00:00.0-00:00" minOccurs="0"
name="reportedDate" type="xs:dateTime"/>
        <xs:element minOccurs="0" name="reportedSource" type="xs:string"/>
        <xs:element minOccurs="0" name="serviceType" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="categorization">
    <xs:sequence>
        <xs:element minOccurs="0" name="tier1" type="xs:string"/>
        <xs:element minOccurs="0" name="tier2" type="xs:string"/>
        <xs:element minOccurs="0" name="tier3" type="xs:string"/>
    </xs:sequence>

```

```

</xs:complexType>
<xs:complexType name="product">
  <xs:complexContent>
    <xs:extension base="tns:categorization">
      <xs:sequence>
        <xs:element minOccurs="0" name="manufacturer" type="xs:string"/>
        <xs:element minOccurs="0" name="modelVersion" type="xs:string"/>
        <xs:element minOccurs="0" name="name" type="xs:string"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="contact">
  <xs:sequence>
    <xs:element minOccurs="0" name="company" type="xs:string"/>
    <xs:element minOccurs="0" name="department" type="xs:string"/>
    <xs:element minOccurs="0" name="emailAddress" type="xs:string"/>
    <xs:element minOccurs="0" name="firstName" type="xs:string"/>
    <xs:element minOccurs="0" name="lastName" type="xs:string"/>
    <xs:element minOccurs="0" name="middleName" type="xs:string"/>
    <xs:element minOccurs="0" name="organization" type="xs:string"/>
    <xs:element minOccurs="0" name="phoneNumber" type="xs:string"/>
    <xs:element minOccurs="0" name="site" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="objectReference">
  <xs:sequence>
    <xs:element minOccurs="0" name="identifier" type="xs:string"/>
    <xs:element minOccurs="0" name="name" type="xs:string"/>
    <xs:element minOccurs="0" name="role" type="xs:string"/>
    <xs:element minOccurs="0" name="type" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="resolution">
  <xs:complexContent>
    <xs:extension base="tns:categorization">
      <xs:sequence>
        <xs:element minOccurs="0" name="cause" type="xs:string"/>
        <xs:element minOccurs="0" name="closureSource" type="xs:string"/>
        <xs:element minOccurs="0" name="detail" type="xs:string"/>
        <xs:element default="1970-01-01T00:00:00.0-00:00" minOccurs="0"
name="forecastResolutionDate" type="xs:dateTime"/>
        <xs:element minOccurs="0" name="method" type="xs:string"/>
        <xs:element minOccurs="0" name="product" type="tns:product"/>
        <xs:element default="1970-01-01T00:00:00.0-00:00" minOccurs="0"
name="requiredDate" type="xs:dateTime"/>
        <xs:element default="1970-01-01T00:00:00.0-00:00" minOccurs="0"
name="resolutionDate" type="xs:dateTime"/>
        <xs:element minOccurs="0" name="satisfactionRating" type="xs:string"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="vendor">
  <xs:sequence>
    <xs:element minOccurs="0" name="assignmentStatus" type="xs:string"/>
    <xs:element minOccurs="0" name="contact" type="tns:contact"/>
    <xs:element minOccurs="0" name="identifier" type="xs:string"/>
    <xs:element default="1970-01-01T00:00:00.0-00:00" minOccurs="0"
name="reportedToVendorDate" type="xs:dateTime"/>
    <xs:element default="1970-01-01T00:00:00.0-00:00" minOccurs="0"

```

```

name="resolvedDate" type="xs:dateTime"/>
  <xs:element default="1970-01-01T00:00:00.0-00:00" minOccurs="0"
name="respondedToVendorDate" type="xs:dateTime"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ci">
  <xs:complexContent>
    <xs:extension base="tns:itObject">
      <xs:sequence>
        <xs:element minOccurs="0" name="assignee" type="xs:string"/>
        <xs:element minOccurs="0" name="category" type="xs:string"/>
        <xs:element minOccurs="0" name="className" type="xs:string"/>
        <xs:element minOccurs="0" name="classNamespace" type="xs:string"/>
        <xs:element minOccurs="0" name="compliance" type="tns:compliance"/>
        <xs:element default="1970-01-01T00:00:00.0-00:00" minOccurs="0"
name="createDate" type="xs:dateTime"/>
        <xs:element minOccurs="0" name="datasetId" type="xs:string"/>
        <xs:element minOccurs="0" name="description" type="xs:string"/>
        <xs:element minOccurs="0" name="hostName" type="xs:string"/>
        <xs:element minOccurs="0" name="item" type="xs:string"/>
        <xs:element default="1970-01-01T00:00:00.0-00:00" minOccurs="0"
name="lastModified" type="xs:dateTime"/>
        <xs:element minOccurs="0" name="lastModifiedBy" type="xs:string"/>
        <xs:element minOccurs="0" name="manufacturer" type="xs:string"/>
        <xs:element minOccurs="0" name="model" type="xs:string"/>
        <xs:element minOccurs="0" name="name" type="xs:string"/>
        <xs:element minOccurs="0" name="relationships">
          <xs:complexType>
            <xs:sequence>
              <xs:element maxOccurs="unbounded" minOccurs="0"
name="relationship" type="tns:ciRelationship"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element minOccurs="0" name="serialNumber" type="xs:string"/>
        <xs:element minOccurs="0" name="status" type="xs:string"/>
        <xs:element minOccurs="0" name="submitter" type="xs:string"/>
        <xs:element minOccurs="0" name="summary" type="xs:string"/>
        <xs:element minOccurs="0" name="type" type="xs:string"/>
        <xs:element minOccurs="0" name="version" type="xs:string"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="compliance">
  <xs:sequence>
    <xs:element minOccurs="0" name="policyIdentifier" type="xs:string"/>
    <xs:element minOccurs="0" name="policyName" type="xs:string"/>
    <xs:element minOccurs="0" name="status" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ciRelationship">
  <xs:sequence>
    <xs:element minOccurs="0" name="className" type="xs:string"/>
    <xs:element minOccurs="0" name="classNamespace" type="xs:string"/>
    <xs:element minOccurs="0" name="datasetId" type="xs:string"/>
    <xs:element minOccurs="0" name="identifier" type="xs:string"/>
    <xs:element minOccurs="0" name="name" type="xs:string"/>
    <xs:element minOccurs="0" name="sourceRole" type="xs:string"/>
    <xs:element minOccurs="0" name="targets">
      <xs:complexType>

```

```

        <xs:sequence>
          <xs:element maxOccurs="unbounded" minOccurs="0" name="target"
type="tns:objectReference"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="change">
  <xs:complexContent>
    <xs:extension base="tns:itObject">
      <xs:sequence>
        <xs:element default="1970-01-01T00:00:00.0-00:00" minOccurs="0"
name="actualEnd" type="xs:dateTime"/>
        <xs:element default="1970-01-01T00:00:00.0-00:00" minOccurs="0"
name="actualStart" type="xs:dateTime"/>
        <xs:element minOccurs="0" name="businessJustification" type="xs:string"/>
        <xs:element minOccurs="0" name="categorization"
type="tns:categorization"/>
        <xs:element minOccurs="0" name="company" type="xs:string"/>
        <xs:element minOccurs="0" name="createDate" type="xs:dateTime"/>
        <xs:element minOccurs="0" name="description" type="xs:string"/>
        <xs:element minOccurs="0" name="impact" type="xs:string"/>
        <xs:element default="1970-01-01T00:00:00.0-00:00" minOccurs="0"
name="lastModified" type="xs:dateTime"/>
        <xs:element minOccurs="0" name="lastModifiedBy" type="xs:string"/>
        <xs:element minOccurs="0" name="logs">
          <xs:complexType>
            <xs:sequence>
              <xs:element maxOccurs="unbounded" minOccurs="0" name="log"
type="tns:objectReference"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element minOccurs="0" name="priority" type="xs:string"/>
        <xs:element minOccurs="0" name="product" type="tns:product"/>
        <xs:element minOccurs="0" name="reason" type="xs:string"/>
        <xs:element minOccurs="0" name="relationships">
          <xs:complexType>
            <xs:sequence>
              <xs:element maxOccurs="unbounded" minOccurs="0"
name="relationship" type="tns:objectReference"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element minOccurs="0" name="requester" type="tns:contact"/>
        <xs:element minOccurs="0" name="riskLevel" type="xs:string"/>
        <xs:element default="1970-01-01T00:00:00.0-00:00" minOccurs="0"
name="scheduledEnd" type="xs:dateTime"/>
        <xs:element default="1970-01-01T00:00:00.0-00:00" minOccurs="0"
name="scheduledStart" type="xs:dateTime"/>
        <xs:element minOccurs="0" name="status" type="xs:string"/>
        <xs:element minOccurs="0" name="summary" type="xs:string"/>
        <xs:element minOccurs="0" name="timing" type="xs:string"/>
        <xs:element minOccurs="0" name="type" type="xs:string"/>
        <xs:element minOccurs="0" name="urgency" type="xs:string"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="problem">

```

```

    <xs:complexContent>
      <xs:extension base="tns:itObject">
        <xs:sequence>
          <xs:element minOccurs="0" name="categorization"
type="tns:categorization"/>
          <xs:element minOccurs="0" name="company" type="xs:string"/>
          <xs:element default="1970-01-01T00:00:00.0-00:00" minOccurs="0"
name="createDate" type="xs:dateTime"/>
          <xs:element minOccurs="0" name="description" type="xs:string"/>
          <xs:element minOccurs="0" name="externalProblemIdentifier"
type="xs:string"/>
          <xs:element minOccurs="0" name="impact" type="xs:string"/>
          <xs:element minOccurs="0" name="investigationDriver"
type="xs:string"/>
          <xs:element default="1970-01-01T00:00:00.0-00:00" minOccurs="0"
name="lastModified" type="xs:dateTime"/>
          <xs:element minOccurs="0" name="lastModifiedBy" type="xs:string"/>
          <xs:element minOccurs="0" name="logs">
            <xs:complexType>
              <xs:sequence>
                <xs:element maxOccurs="unbounded" minOccurs="0" name="log"
type="tns:objectReference"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element minOccurs="0" name="priority" type="xs:string"/>
          <xs:element minOccurs="0" name="product" type="tns:product"/>
          <xs:element minOccurs="0" name="relationships">
            <xs:complexType>
              <xs:sequence>
                <xs:element maxOccurs="unbounded" minOccurs="0"
name="relationship" type="tns:objectReference"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element minOccurs="0" name="requester" type="tns:contact"/>
          <xs:element minOccurs="0" name="rootCause" type="xs:string"/>
          <xs:element minOccurs="0" name="severity" type="xs:string"/>
          <xs:element minOccurs="0" name="status" type="xs:string"/>
          <xs:element minOccurs="0" name="summary" type="xs:string"/>
          <xs:element default="1970-01-01T00:00:00.0-00:00" minOccurs="0"
name="targetResolution" type="xs:dateTime"/>
          <xs:element minOccurs="0" name="urgency" type="xs:string"/>
          <xs:element minOccurs="0" name="weight" type="xs:string"/>
          <xs:element minOccurs="0" name="workaround" type="xs:string"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="data">
    <xs:sequence>
      <xs:choice maxOccurs="unbounded" minOccurs="0">
        <xs:element name="string-field" type="tns:stringField"/>
        <xs:element name="date-field" type="tns:dateField"/>
        <xs:element name="number-field" type="tns:numberField"/>
        <xs:element name="diary-list-field" type="tns:diaryListField"/>
        <xs:element name="complex-field" type="tns:complexField"/>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="UnsupportedOperationException"

```

```

type="tns:UnsupportedOperationException"/>
  <xs:complexType name="UnsupportedOperationException">
    <xs:sequence/>
  </xs:complexType>
  <xs:element name="OperationsException" type="tns:OperationsException"/>
  <xs:complexType name="OperationsException">
    <xs:sequence/>
  </xs:complexType>
  <xs:element name="ObjectNotFoundException" type="tns:ObjectNotFoundException"/>
  <xs:complexType name="ObjectNotFoundException">
    <xs:sequence/>
  </xs:complexType>
  <xs:element name="NonRecoverableOperationsException"
type="tns:NonRecoverableOperationsException"/>
  <xs:complexType name="NonRecoverableOperationsException">
    <xs:sequence/>
  </xs:complexType>
</xs:schema>
</wsdl:types>
<wsdl:message name="getResponse">
  <wsdl:part element="tns:getResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="create">
  <wsdl:part element="tns:create" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="clearResponse">
  <wsdl:part element="tns:clearResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="NonRecoverableOperationsException">
  <wsdl:part element="tns:NonRecoverableOperationsException"
name="NonRecoverableOperationsException"></wsdl:part>
</wsdl:message>
<wsdl:message name="acknowledgeResponse">
  <wsdl:part element="tns:acknowledgeResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="update">
  <wsdl:part element="tns:update" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="ObjectNotFoundException">
  <wsdl:part element="tns:ObjectNotFoundException" name="ObjectNotFoundException"></
wsdl:part>
</wsdl:message>
<wsdl:message name="acknowledge">
  <wsdl:part element="tns:acknowledge" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="get">
  <wsdl:part element="tns:get" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="createResponse">
  <wsdl:part element="tns:createResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="findResponse">
  <wsdl:part element="tns:findResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="OperationsException">
  <wsdl:part element="tns:OperationsException" name="OperationsException"></
wsdl:part>
</wsdl:message>
<wsdl:message name="find">
  <wsdl:part element="tns:find" name="parameters"></wsdl:part>
</wsdl:message>

```

```

    <wsdl:message name="clear">
      <wsdl:part element="tns:clear" name="parameters"></wsdl:part>
    </wsdl:message>
    <wsdl:message name="UnsupportedOperationException">
      <wsdl:part element="tns:UnsupportedOperationException"
name="UnsupportedOperationException"></wsdl:part>
    </wsdl:message>
    <wsdl:message name="updateResponse">
      <wsdl:part element="tns:updateResponse" name="parameters"></wsdl:part>
    </wsdl:message>
    <wsdl:portType name="IEventService">
      <wsdl:operation name="clear">
        <wsdl:input message="tns:clear" name="clear"></wsdl:input>
        <wsdl:output message="tns:clearResponse" name="clearResponse"></
wsdl:output>
        <wsdl:fault message="tns:ObjectNotFoundException"
name="ObjectNotFoundException"></wsdl:fault>
        <wsdl:fault message="tns:UnsupportedOperationException"
name="UnsupportedOperationException"></wsdl:fault>
        <wsdl:fault message="tns:OperationsException" name="OperationsException"></
wsdl:fault>
        <wsdl:fault message="tns:NonRecoverableOperationsException"
name="NonRecoverableOperationsException"></wsdl:fault>
      </wsdl:operation>
      <wsdl:operation name="update">
        <wsdl:input message="tns:update" name="update"></wsdl:input>
        <wsdl:output message="tns:updateResponse" name="updateResponse"></
wsdl:output>
        <wsdl:fault message="tns:ObjectNotFoundException"
name="ObjectNotFoundException"></wsdl:fault>
        <wsdl:fault message="tns:OperationsException" name="OperationsException"></
wsdl:fault>
        <wsdl:fault message="tns:NonRecoverableOperationsException"
name="NonRecoverableOperationsException"></wsdl:fault>
      </wsdl:operation>
      <wsdl:operation name="acknowledge">
        <wsdl:input message="tns:acknowledge" name="acknowledge"></wsdl:input>
        <wsdl:output message="tns:acknowledgeResponse"
name="acknowledgeResponse"></wsdl:output>
        <wsdl:fault message="tns:ObjectNotFoundException"
name="ObjectNotFoundException"></wsdl:fault>
        <wsdl:fault message="tns:UnsupportedOperationException"
name="UnsupportedOperationException"></wsdl:fault>
        <wsdl:fault message="tns:OperationsException" name="OperationsException"></
wsdl:fault>
        <wsdl:fault message="tns:NonRecoverableOperationsException"
name="NonRecoverableOperationsException"></wsdl:fault>
      </wsdl:operation>
      <wsdl:operation name="get">
        <wsdl:input message="tns:get" name="get"></wsdl:input>
        <wsdl:output message="tns:getResponse" name="getResponse"></wsdl:output>
        <wsdl:fault message="tns:ObjectNotFoundException"
name="ObjectNotFoundException"></wsdl:fault>
        <wsdl:fault message="tns:UnsupportedOperationException"
name="UnsupportedOperationException"></wsdl:fault>
        <wsdl:fault message="tns:OperationsException" name="OperationsException"></
wsdl:fault>
        <wsdl:fault message="tns:NonRecoverableOperationsException"
name="NonRecoverableOperationsException"></wsdl:fault>
      </wsdl:operation>
      <wsdl:operation name="find">

```



```

        <wsdl:input message="tns:find" name="find"></wsdl:input>
        <wsdl:output message="tns:findResponse" name="findResponse"></wsdl:output>
        <wsdl:fault message="tns:UnsupportedOperationException"
name="UnsupportedOperationException"></wsdl:fault>
        <wsdl:fault message="tns:OperationsException" name="OperationsException"></
wsdl:fault>
        <wsdl:fault message="tns:NonRecoverableOperationsException"
name="NonRecoverableOperationsException"></wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="create">
        <wsdl:input message="tns:create" name="create"></wsdl:input>
        <wsdl:output message="tns:createResponse" name="createResponse"></wsdl:output>
        <wsdl:fault message="tns:OperationsException" name="OperationsException"></
wsdl:fault>
        <wsdl:fault message="tns:NonRecoverableOperationsException"
name="NonRecoverableOperationsException"></wsdl:fault>
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="EventServiceSoapBinding" type="tns:IEventService">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="clear">
        <soap:operation soapAction="" style="document"/>
        <wsdl:input name="clear">
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output name="clearResponse">
            <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="ObjectNotFoundException">
            <soap:fault name="ObjectNotFoundException" use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="UnsupportedOperationException">
            <soap:fault name="UnsupportedOperationException" use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="OperationsException">
            <soap:fault name="OperationsException" use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="NonRecoverableOperationsException">
            <soap:fault name="NonRecoverableOperationsException" use="literal"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="update">
        <soap:operation soapAction="" style="document"/>
        <wsdl:input name="update">
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output name="updateResponse">
            <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="ObjectNotFoundException">
            <soap:fault name="ObjectNotFoundException" use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="OperationsException">
            <soap:fault name="OperationsException" use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="NonRecoverableOperationsException">
            <soap:fault name="NonRecoverableOperationsException" use="literal"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="acknowledge">
        <soap:operation soapAction="" style="document"/>

```

```

<wsdl:input name="acknowledge">
  <soap:body use="literal"/>
</wsdl:input>
<wsdl:output name="acknowledgeResponse">
  <soap:body use="literal"/>
</wsdl:output>
<wsdl:fault name="ObjectNotFoundException">
  <soap:fault name="ObjectNotFoundException" use="literal"/>
</wsdl:fault>
<wsdl:fault name="UnsupportedOperationException">
  <soap:fault name="UnsupportedOperationException" use="literal"/>
</wsdl:fault>
<wsdl:fault name="OperationsException">
  <soap:fault name="OperationsException" use="literal"/>
</wsdl:fault>
<wsdl:fault name="NonRecoverableOperationsException">
  <soap:fault name="NonRecoverableOperationsException" use="literal"/>
</wsdl:fault>
</wsdl:operation>
<wsdl:operation name="get">
  <soap:operation soapAction="" style="document"/>
  <wsdl:input name="get">
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output name="getResponse">
    <soap:body use="literal"/>
  </wsdl:output>
  <wsdl:fault name="ObjectNotFoundException">
    <soap:fault name="ObjectNotFoundException" use="literal"/>
  </wsdl:fault>
  <wsdl:fault name="UnsupportedOperationException">
    <soap:fault name="UnsupportedOperationException" use="literal"/>
  </wsdl:fault>
  <wsdl:fault name="OperationsException">
    <soap:fault name="OperationsException" use="literal"/>
  </wsdl:fault>
  <wsdl:fault name="NonRecoverableOperationsException">
    <soap:fault name="NonRecoverableOperationsException" use="literal"/>
  </wsdl:fault>
</wsdl:operation>
<wsdl:operation name="find">
  <soap:operation soapAction="" style="document"/>
  <wsdl:input name="find">
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output name="findResponse">
    <soap:body use="literal"/>
  </wsdl:output>
  <wsdl:fault name="UnsupportedOperationException">
    <soap:fault name="UnsupportedOperationException" use="literal"/>
  </wsdl:fault>
  <wsdl:fault name="OperationsException">
    <soap:fault name="OperationsException" use="literal"/>
  </wsdl:fault>
  <wsdl:fault name="NonRecoverableOperationsException">
    <soap:fault name="NonRecoverableOperationsException" use="literal"/>
  </wsdl:fault>
</wsdl:operation>
<wsdl:operation name="create">
  <soap:operation soapAction="" style="document"/>
  <wsdl:input name="create">

```

```
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="createResponse">
        <soap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="OperationsException">
        <soap:fault name="OperationsException" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="NonRecoverableOperationsException">
        <soap:fault name="NonRecoverableOperationsException" use="literal"/>
    </wsdl:fault>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="EventService">
    <wsdl:port binding="tns:EventServiceSoapBinding" name="EventServicePort">
        <soap:address location="https://localhost:8443/services/omnibus/EventService"/>
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

D

Enterprise Manager Event Examples

This appendix contains examples that show the XML format of events from Enterprise Manager that conform to the schemas defined in [Enterprise Manager Event Format](#). You can use these examples to help identify the fields to map when customizing the event templates. You can also use them to test your customizations after they have been built.

The following examples are presented:

- [Create Example](#)
- [Update Example](#)
- [Clear Example](#)

Create Example

The following example shows the XML format Enterprise Manager generates for Netcool/OMNIbus create operations.

```
<EMEvent xmlns="http://xmlns.oracle.com/sysman/connector">
  <ConnectorGUID>F4666F4BE15F464995749B6984DD276D</ConnectorGUID>
  <NotificationRuleOwner>SYSMAN</NotificationRuleOwner>
  <NotificationRuleName>Netcool/OMNIbus Rule</NotificationRuleName>
  <Property>
    <Name>Notification_Method_Name</Name>
    <Value>Netcool/OMNIbus</Value>
  </Property>
  <Property>
    <Name>AuthenticationType</Name>
    <Value>HTTPBasicAuthentication</Value>
  </Property>
  <Property>
    <Name>IsNewTargetType</Name>
    <Value>yes</Value>
  </Property>
  <SystemAttributes>
    <EventClass>metric_alert</EventClass>
    <EventID>C14525FD49067F04E040020A780113B0</EventID>
    <SequenceID>C14525FD49077F04E040020A780113B0</SequenceID>
    <ReportedDate>2012-06-07T14:48:55.000-05:00</ReportedDate>
    <DisplayTZ>America/Chicago</DisplayTZ>
    <EventName>Load:memUsedPct</EventName>
    <Severity>Critical</Severity>
    <SeverityCode>CRITICAL</SeverityCode>
    <SourceInfo>
      <SourceObjInfo>
        <ObjID>D93A2E4E1622013D74D9534ECB090944</ObjID>
        <ObjName>orcsvr1.iwavesoftware.com</ObjName>
        <ObjOwner>SYSMAN</ObjOwner>
        <SourceObjType>TARGET</SourceObjType>
        <SourceObjSubType>host</SourceObjSubType>
      </SourceObjInfo>
      <TargetInfo>
```

```
<TargetGUID>D93A2E4E1622013D74D9534ECB090944</TargetGUID>
<TargetName>orcsvr1.iwavesoftware.com</TargetName>
<TargetType>host</TargetType>
<TargetTypeLabel>Host</TargetTypeLabel>
<TargetURL>https://orcsvr1.iwavesoftware.com:7801/em/redirect?
pageType=TARGET_HOMEPAGE&amp;targetName=orcsvr1.iwavesoftware.com&amp;targetType=
host</TargetURL>
  <TargetProperty>
    <Name>Target_Host</Name>
    <Value>orcsvr1.iwavesoftware.com</Value>
  </TargetProperty>
  <TargetProperty>
    <Name>Target Version</Name>
    <Value>5.5.0.0.0</Value>
  </TargetProperty>
  <TargetProperty>
    <Name>Line of Business</Name>
    <Value>Software</Value>
  </TargetProperty>
  <TargetProperty>
    <Name>Location</Name>
    <Value>Frisco, TX</Value>
  </TargetProperty>
  <TargetProperty>
    <Name>Comment</Name>
    <Value>Testing target properties</Value>
  </TargetProperty>
</TargetInfo>
</SourceInfo>
<Message>Memory Utilization is 84.294%, crossed warning (50) or critical
(70) threshold.</Message>
  <EventURL>https://orcsvr1.iwavesoftware.com:7801/em/redirect?pageType=sdk-
core-event-console-detailEvent&amp;issueID=C14525FD49077F04E040020A780113B0</
EventURL>
  <AutoClose>>true</AutoClose>
  <EventCategory>Capacity</EventCategory>
</SystemAttributes>
<EventClassSpecificAttributes>
  <StringAttribute>
    <Name>is_thresholdable</Name>
    <Value>1</Value>
  </StringAttribute>
  <StringAttribute>
    <Name>coll_name</Name>
    <Value>LoadLinux</Value>
  </StringAttribute>
  <StringAttribute>
    <Name>is_metric_extension</Name>
    <Value>0</Value>
  </StringAttribute>
  <StringAttribute>
    <Name>metric_column_resbundle</Name>
    <Value>oracle.sysman.eml.rsc.gen.hostMsg</Value>
  </StringAttribute>
  <StringAttribute>
    <Name>metric_description_resbundle</Name>
    <Value>oracle.sysman.eml.rsc.gen.hostMsg</Value>
  </StringAttribute>
  <StringAttribute>
    <Name>unit_resbundle</Name>
    <Value>oracle.sysman.eml.rsc.gen.hostMsg</Value>
  </StringAttribute>
</EventClassSpecificAttributes>
</SystemAttributes>
```

```
</StringAttribute>
<StringAttribute>
  <Name>cycle_guid</Name>
  <Value>C14525FD49037F04E040020A780113B0</Value>
</StringAttribute>
<StringAttribute>
  <Name>is_remote</Name>
  <Value>0</Value>
</StringAttribute>
<StringAttribute>
  <Name>metric_type</Name>
  <Value>0</Value>
</StringAttribute>
<StringAttribute>
  <Name>metric_guid</Name>
  <Value>86821B5F0CE858D6E4A7F7390E88B73C</Value>
</StringAttribute>
<StringAttribute>
  <Name>num_keys</Name>
  <Value>0</Value>
</StringAttribute>
<StringAttribute>
  <Name>key_value</Name>
  <Value></Value>
</StringAttribute>
<StringAttribute>
  <Name>metric_description_nlsid</Name>
</StringAttribute>
<StringAttribute>
  <Name>value</Name>
  <Value>84.294</Value>
</StringAttribute>
<StringAttribute>
  <Name>is_long_running</Name>
  <Value>0</Value>
</StringAttribute>
<StringAttribute>
  <Name>metric_group</Name>
  <Value>Load</Value>
</StringAttribute>
<StringAttribute>
  <Name>is_udm</Name>
  <Value>0</Value>
</StringAttribute>
<StringAttribute>
  <Name>metric_description</Name>
</StringAttribute>
<StringAttribute>
  <Name>metric_column_nlsid</Name>
  <Value>host_load_memUsedPct</Value>
</StringAttribute>
<StringAttribute>
  <Name>metric_column</Name>
  <Value>memUsedPct</Value>
</StringAttribute>
<StringAttribute>
  <Name>unit_nlsid</Name>
  <Value>em_sys_standard_percent</Value>
</StringAttribute>
<StringAttribute>
  <Name>unit</Name>
```

```

    <Value>%</Value>
  </StringAttribute>
  <StringAttribute>
    <Name>metric_group_nlsid</Name>
    <Value>host_load</Value>
  </StringAttribute>
  <StringAttribute>
    <Name>metric_group_resbundle</Name>
    <Value>oracle.sysman.eml.rsc.gen.hostMsg</Value>
  </StringAttribute>
  <StringAttribute>
    <Name>severity_guid</Name>
    <Value>C14525FD49037F04E040020A780113B0</Value>
  </StringAttribute>
</EventClassSpecificAttributes>
</EMEvent>

```

Update Example

The following example shows the XML format Enterprise Manager generates for Netcool/OMNIbus update operations.

```

<EMEvent xmlns="http://xmlns.oracle.com/sysman/connector">
  <ConnectorGUID>F4666F4BE15F464995749B6984DD276D</ConnectorGUID>
  <ExternalEventID>C14525FD49067F04E040020A780113B0</ExternalEventID>
  <NotificationRuleOwner>SYSMAN</NotificationRuleOwner>
  <NotificationRuleName>Netcool/OMNIbus Rule</NotificationRuleName>
  <Property>
    <Name>Notification_Method_Name</Name>
    <Value>Netcool/OMNIbus</Value>
  </Property>
  <Property>
    <Name>AuthenticationType</Name>
    <Value>HTTPBasicAuthentication</Value>
  </Property>
  <Property>
    <Name>IsNewTargetType</Name>
    <Value>yes</Value>
  </Property>
  <SystemAttributes>
    <EventClass>metric_alert</EventClass>
    <EventID>C14525FB87803EB3E040020A780113A7</EventID>
    <SequenceID>C14525FD49077F04E040020A780113B0</SequenceID>
    <ReportedDate>2012-06-07T14:50:03.000-05:00</ReportedDate>
    <DisplayTZ>America/Chicago</DisplayTZ>
    <EventName>Load:memUsedPct</EventName>
    <Severity>Warning</Severity>
    <SeverityCode>WARNING</SeverityCode>
    <SourceInfo>
      <SourceObjInfo>
        <ObjID>D93A2E4E1622013D74D9534ECB090944</ObjID>
        <ObjName>orcsvr1.iwavesoftware.com</ObjName>
        <ObjOwner>SYSMAN</ObjOwner>
        <SourceObjType>TARGET</SourceObjType>
        <SourceObjSubType>host</SourceObjSubType>
      </SourceObjInfo>
      <TargetInfo>
        <TargetGUID>D93A2E4E1622013D74D9534ECB090944</TargetGUID>
        <TargetName>orcsvr1.iwavesoftware.com</TargetName>
        <TargetType>host</TargetType>
      </TargetInfo>
    </SourceInfo>
  </SystemAttributes>
</EMEvent>

```

```
<TargetTypeLabel>Host</TargetTypeLabel>
<TargetURL>https://orcsvr1.iwavesoftware.com:7801/em/redirect?
pageType=TARGET_HOMEPAGE&targetName=orcsvr1.iwavesoftware.com&targetType=host</
TargetURL>
<TargetProperty>
  <Name>Target_Host</Name>
  <Value>orcsvr1.iwavesoftware.com</Value>
</TargetProperty>
<TargetProperty>
  <Name>Target Version</Name>
  <Value>5.5.0.0.0</Value>
</TargetProperty>
<TargetProperty>
  <Name>Line of Business</Name>
  <Value>Software</Value>
</TargetProperty>
<TargetProperty>
  <Name>Location</Name>
  <Value>Frisco, TX</Value>
</TargetProperty>
<TargetProperty>
  <Name>Comment</Name>
  <Value>Testing target properties</Value>
</TargetProperty>
</TargetInfo>
</SourceInfo>
<Message>Memory Utilization is 84.572%, crossed warning (50) or critical (95)
threshold.</Message>
<EventURL>https://orcsvr1.iwavesoftware.com:7801/em/redirect?pageType=sdk-core-
event-console-detailEvent&issueID=C14525FD49077F04E040020A780113B0</EventURL>
<AutoClose>>true</AutoClose>
<EventCategory>Capacity</EventCategory>
</SystemAttributes>
<EventClassSpecificAttributes>
<StringAttribute>
  <Name>is_thresholdable</Name>
  <Value>1</Value>
</StringAttribute>
<StringAttribute>
  <Name>coll_name</Name>
  <Value>LoadLinux</Value>
</StringAttribute>
<StringAttribute>
  <Name>is_metric_extension</Name>
  <Value>0</Value>
</StringAttribute>
<StringAttribute>
  <Name>metric_column_resbundle</Name>
  <Value>oracle.sysman.eml.rsc.gen.hostMsg</Value>
</StringAttribute>
<StringAttribute>
  <Name>metric_description_resbundle</Name>
  <Value>oracle.sysman.eml.rsc.gen.hostMsg</Value>
</StringAttribute>
<StringAttribute>
  <Name>unit_resbundle</Name>
  <Value>oracle.sysman.eml.rsc.gen.hostMsg</Value>
</StringAttribute>
<StringAttribute>
  <Name>cycle_guid</Name>
  <Value>C14525FD49037F04E040020A780113B0</Value>
```



```
</StringAttribute>
<StringAttribute>
  <Name>is_remote</Name>
  <Value>0</Value>
</StringAttribute>
<StringAttribute>
  <Name>metric_type</Name>
  <Value>0</Value>
</StringAttribute>
<StringAttribute>
  <Name>metric_guid</Name>
  <Value>86821B5F0CE858D6E4A7F7390E88B73C</Value>
</StringAttribute>
<StringAttribute>
  <Name>num_keys</Name>
  <Value>0</Value>
</StringAttribute>
<StringAttribute>
  <Name>key_value</Name>
  <Value></Value>
</StringAttribute>
<StringAttribute>
  <Name>metric_description_nlsid</Name>
</StringAttribute>
<StringAttribute>
  <Name>value</Name>
  <Value>84.572</Value>
</StringAttribute>
<StringAttribute>
  <Name>is_long_running</Name>
  <Value>0</Value>
</StringAttribute>
<StringAttribute>
  <Name>metric_group</Name>
  <Value>Load</Value>
</StringAttribute>
<StringAttribute>
  <Name>is_udm</Name>
  <Value>0</Value>
</StringAttribute>
<StringAttribute>
  <Name>metric_description</Name>
</StringAttribute>
<StringAttribute>
  <Name>metric_column_nlsid</Name>
  <Value>host_load_memUsedPct</Value>
</StringAttribute>
<StringAttribute>
  <Name>metric_column</Name>
  <Value>memUsedPct</Value>
</StringAttribute>
<StringAttribute>
  <Name>unit_nlsid</Name>
  <Value>em_sys_standard_percent</Value>
</StringAttribute>
<StringAttribute>
  <Name>unit</Name>
  <Value>%</Value>
</StringAttribute>
<StringAttribute>
  <Name>metric_group_nlsid</Name>
```

```

    <Value>host_load</Value>
  </StringAttribute>
  <StringAttribute>
    <Name>metric_group_resbundle</Name>
    <Value>oracle.sysman.eml.rsc.gen.hostMsg</Value>
  </StringAttribute>
  <StringAttribute>
    <Name>severity_guid</Name>
    <Value>C14525FB877D3EB3E040020A780113A7</Value>
  </StringAttribute>
</EventClassSpecificAttributes>
</EMEvent>

```

Clear Example

The following example shows the XML format Enterprise Manager generates for Netcool/OMNIBus clear operations.

```

<EMEvent xmlns="http://xmlns.oracle.com/sysman/connector">
  <ConnectorGUID>F4666F4BE15F464995749B6984DD276D</ConnectorGUID>
  <ExternalEventID>C14525FD49067F04E040020A780113B0</ExternalEventID>
  <NotificationRuleOwner>SYSMAN</NotificationRuleOwner>
  <NotificationRuleName>Netcool/OMNIBus Rule</NotificationRuleName>
  <Property>
    <Name>Notification_Method_Name</Name>
    <Value>Netcool/OMNIBus</Value>
  </Property>
  <Property>
    <Name>AuthenticationType</Name>
    <Value>HTTPBasicAuthentication</Value>
  </Property>
  <Property>
    <Name>IsNewTargetType</Name>
    <Value>yes</Value>
  </Property>
  <SystemAttributes>
    <EventClass>metric_alert</EventClass>
    <EventID>C14525FD490B7F04E040020A780113B0</EventID>
    <SequenceID>C14525FD49077F04E040020A780113B0</SequenceID>
    <ReportedDate>2012-06-07T14:50:35.000-05:00</ReportedDate>
    <DisplayTZ>America/Chicago</DisplayTZ>
    <EventName>Load:memUsedPct</EventName>
    <Severity>Clear</Severity>
    <SeverityCode>CLEAR</SeverityCode>
    <SourceInfo>
      <SourceObjInfo>
        <ObjID>D93A2E4E1622013D74D9534ECB090944</ObjID>
        <ObjName>orcsvr1.iwavesoftware.com</ObjName>
        <ObjOwner>SYSMAN</ObjOwner>
        <SourceObjType>TARGET</SourceObjType>
        <SourceObjSubType>host</SourceObjSubType>
      </SourceObjInfo>
      <TargetInfo>
        <TargetGUID>D93A2E4E1622013D74D9534ECB090944</TargetGUID>
        <TargetName>orcsvr1.iwavesoftware.com</TargetName>
        <TargetType>host</TargetType>
        <TargetTypeLabel>Host</TargetTypeLabel>
        <TargetURL>https://orcsvr1.iwavesoftware.com:7801/em/redirect?
pageType=TARGET_HOMEPAGE&targetName=orcsvr1.iwavesoftware.com&targetType=host</
TargetURL>

```

```
<TargetProperty>
  <Name>Target_Host</Name>
  <Value>orcsvr1.iwavesoftware.com</Value>
</TargetProperty>
<TargetProperty>
  <Name>Target Version</Name>
  <Value>5.5.0.0.0</Value>
</TargetProperty>
<TargetProperty>
  <Name>Line of Business</Name>
  <Value>Software</Value>
</TargetProperty>
<TargetProperty>
  <Name>Location</Name>
  <Value>Frisco, TX</Value>
</TargetProperty>
<TargetProperty>
  <Name>Comment</Name>
  <Value>Testing target properties</Value>
</TargetProperty>
</TargetInfo>
</SourceInfo>
<Message>Memory Utilization is 84.763%, fallen below warning (98) and
critical (99) thresholds.</Message>
<EventURL>https://orcsvr1.iwavesoftware.com:7801/em/redirect?pageType=sdk-
core-event-console-detailEvent&amp;issueID=C14525FD49077F04E040020A780113B0</
EventURL>
  <AutoClose>true</AutoClose>
  <EventCategory>Capacity</EventCategory>
</SystemAttributes>
<EventClassSpecificAttributes>
  <StringAttribute>
    <Name>is_thresholdable</Name>
    <Value>1</Value>
  </StringAttribute>
  <StringAttribute>
    <Name>coll_name</Name>
    <Value>LoadLinux</Value>
  </StringAttribute>
  <StringAttribute>
    <Name>is_metric_extension</Name>
    <Value>0</Value>
  </StringAttribute>
  <StringAttribute>
    <Name>metric_column_resbundle</Name>
    <Value>oracle.sysman.eml.rsc.gen.hostMsg</Value>
  </StringAttribute>
  <StringAttribute>
    <Name>metric_description_resbundle</Name>
    <Value>oracle.sysman.eml.rsc.gen.hostMsg</Value>
  </StringAttribute>
  <StringAttribute>
    <Name>unit_resbundle</Name>
    <Value>oracle.sysman.eml.rsc.gen.hostMsg</Value>
  </StringAttribute>
  <StringAttribute>
    <Name>cycle_guid</Name>
    <Value>C14525FD49037F04E040020A780113B0</Value>
  </StringAttribute>
  <StringAttribute>
    <Name>is_remote</Name>
```

```
<Value>0</Value>
</StringAttribute>
<StringAttribute>
  <Name>metric_type</Name>
  <Value>0</Value>
</StringAttribute>
<StringAttribute>
  <Name>metric_guid</Name>
  <Value>86821B5F0CE858D6E4A7F7390E88B73C</Value>
</StringAttribute>
<StringAttribute>
  <Name>num_keys</Name>
  <Value>0</Value>
</StringAttribute>
<StringAttribute>
  <Name>key_value</Name>
  <Value></Value>
</StringAttribute>
<StringAttribute>
  <Name>metric_description_nlsid</Name>
</StringAttribute>
<StringAttribute>
  <Name>value</Name>
  <Value>84.763</Value>
</StringAttribute>
<StringAttribute>
  <Name>is_long_running</Name>
  <Value>0</Value>
</StringAttribute>
<StringAttribute>
  <Name>metric_group</Name>
  <Value>Load</Value>
</StringAttribute>
<StringAttribute>
  <Name>is_udm</Name>
  <Value>0</Value>
</StringAttribute>
<StringAttribute>
  <Name>metric_description</Name>
</StringAttribute>
<StringAttribute>
  <Name>metric_column_nlsid</Name>
  <Value>host_load_memUsedPct</Value>
</StringAttribute>
<StringAttribute>
  <Name>metric_column</Name>
  <Value>memUsedPct</Value>
</StringAttribute>
<StringAttribute>
  <Name>unit_nlsid</Name>
  <Value>em_sys_standard_percent</Value>
</StringAttribute>
<StringAttribute>
  <Name>unit</Name>
  <Value>%</Value>
</StringAttribute>
<StringAttribute>
  <Name>metric_group_nlsid</Name>
  <Value>host_load</Value>
</StringAttribute>
<StringAttribute>
```

```
    <Name>metric_group_resbundle</Name>
    <Value>oracle.sysman.eml.rsc.gen.hostMsg</Value>
  </StringAttribute>
  <StringAttribute>
    <Name>severity_guid</Name>
    <Value>C14525FD49097F04E040020A780113B0</Value>
  </StringAttribute>
</EventClassSpecificAttributes>
</EMEvent>
```